UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

DEPARTAMENTO DE TEORÍA DE LA SEÑAL Y COMUNICACIONES

INGENIERÍA TÉCNICA DE TELECOMUNICACIÓN SISTEMAS DE TELECOMUNICACIÓN

PREDICCIÓN DE INTENTOS DE SUICIDIO

PROYECTO FIN DE CARRERA

LEGANÉS. DICIEMBRE 2013

AUTOR: TAMARA BOUZA LÓPEZ

TUTOR: Dr. FERNANDO PÉREZ CRUZ

Título: Predicción de intentos de suicidio

Autor: Tamara Bouza López

Tutor: Dr. Fernando Pérez Cruz

La defensa del presente Proyecto Fin de Carrera se realizó el día 11 de Diciembre de 2013, siendo calificado por el tribunal

PRESIDENTE

SECRETARIO

VOCAL

Habiendo obtenido la siguiente calificación

Presidente Secretario Vocal

AGRADECIMIENTOS

Llegando al fin de una etapa importante en mi vida, no puedo dejar de agradecer a mis padres, Ángeles y José, su amor y apoyo incondicional en este y en todos los momentos de mi vida. Sin ellos no sería quien soy y no habría llegado hasta aquí. Gracias por todo lo que me habéis enseñado y por ser los mejores padres del mundo.

A mi hermana, Noemí, por estar ahí siempre que la he necesitado y porque su afán por aprender y no rendirse nunca, me han motivado a lo largo de mi vida.

A mi abuelo Leonardo, por mantenerme unida a todo aquello que me hacía feliz cuando era niña. Eres el mejor abuelo del mundo.

A mi abuela Manuela, por tus cuentos, por tus canciones, por tus sonrisas y porque desde que no estás no he dejado de echarte de menos. Y a mi abuela María, por hacerme reír como nadie.

A Álex, porque todo es mejor desde que estás a mi lado, no tengo palabras para darte las gracias por estar siempre que te he necesitado, por lo mucho que he aprendido y sigo aprendiendo de ti y porque contigo soy la persona más feliz del mundo.

A mis tíos, y en especial a Javi, por ser genial y sacarme una sonrisa en cada momento.

A mis amigas, Arantxa y Dris, porque sois las mejores amigas que alguien puede tener.

Por supuesto, a mi tutor Fernando, por su paciencia y por su ayuda siempre que se la he pedido.

Y a todos los que en algún momento, habéis formado parte de mi vida.

A todos, muchas gracias.

RESUMEN

En este Proyecto de Fin de Carrera, se expone una propuesta para detectar con la menor probabilidad de No Detección posible, si un sujeto va a intentar suicidarse.

Para ello, partiendo de las respuestas obtenidas en una encuesta realizada a un conjunto representativo de la población estadounidense, se aplicarán los conceptos de *Aprendizaje Máquina Supervisado,* el algoritmo de clasificación de *Naïve Bayes* y el algoritmo de *Validación Cruzada*, entre otros, con el fin de llegar a un clasificador que permita detectar si un paciente es propenso al suicidio y tratarlo de forma adecuada para evitar que llegue a cometerlo.

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

Introducción

I.1 Motivación del proyecto

Actualmente, no es difícil encontrar personas con algún tipo de enfermedad mental: desde una depresión puntual por estar atravesando un momento difícil, hasta enfermedades crónicas de distinta gravedad. Todas ellas tienen una sintomatología y un origen diferente y en general, son diagnosticadas por un psiquíatra o especialista, aunque los estudios en el uso de técnicas de procesamiento digital aplicado sobre imágenes médicas, entre otros, hacen que cada vez sea más frecuente recurrir a la tecnología para el diagnóstico y comprensión de estas enfermedades, a veces tan complejas y difíciles de tratar.

Además, estas enfermedades, dependiendo de la gravedad y de la persona que las sufre, llevan asociadas situaciones de desesperación, agravadas muchas veces por determinados hábitos, y que pueden desembocar en un intento de suicidio por parte de quien las sufre.

Entender por qué se origina un trastorno mental, por qué unas personas son más propensas que otras a padecerlo y tratarlas de forma adecuada es algo muy complejo, y más complejo es aún predecir el comportamiento del paciente para poder aplicarle el tratamiento y/o terapia adecuados y evitar que pueda llegar a hacer daño a alguien o a sí mismo.

Este proyecto, se centrará en la ayuda a la resolución de este problema, es decir, en intentar predecir el comportamiento de una persona en base a sus hábitos y experiencias vividas, para saber si puede llegar a intentar suicidarse y que reciba el tratamiento adecuado para evitarlo.

I.2 Objetivos

El objetivo de este proyecto es encontrar el mejor clasificador que permita decidir con la menor probabilidad de No Detección posible, si una persona va a intentar suicidarse o no. En ningún caso se trata de un proyecto donde se pretenda evitar que una persona se suicide, ya que si alguien quiere quitarse la vida, lo va a hacer. Lo que se quiere realizar, es una herramienta de ayuda para el diagnóstico y posterior tratamiento de aquellas personas cuya intención no es la de morir, pero que debido al problema que sufren, en algún momento llegan a intentarlo, aunque solo sea para llamar la atención.

Para ello, se han analizado las variables y características obtenidas de un estudio previo entre un grupo de la población estadounidense, la encuesta NESARC (National Epidemiologic Survey on Alcohol and Related Conditions), que será explicado brevemente en el siguiente apartado.

Dado el gran conjunto de datos de los que se dispone (véase Capítulo 2), se hace necesario el uso de algoritmos y definiciones que ayuden a la organización de los mismos, ya que el uso de datos erróneos, redundantes o innecesarios, puede producir un mal resultado.

Para llegar al mejor clasificador, se usarán las técnicas y definiciones del aprendizaje máquina, y más concretamente, del método *Leave-one-out* y del algoritmo de clasificación de Naïve-Bayes, con los que se obtendrán un gran número de modelos con unos determinados valores de probabilidad de Falsa Alarma y No Detección que a su vez, estarán asociados a unas determinadas variables de la encuesta. Estas probabilidades deberán ser analizadas para poder seleccionar el mejor clasificador.

De entre todos estos modelos, interesa elegir el de menor probabilidad de No Detección pero también es conveniente que la probabilidad de Falsa Alarma no sea muy alta por lo que habrá que llegar a un "acuerdo" entre ambos factores.

Será en el Capítulo 3 cuando se haga un análisis de los modelos elegidos como válidos para este proyecto.

I.3 NESARC

NESARC, (National Epidemiologic Survey on Alcohol and Related Conditions[1]), es la mayor y más completa encuesta realizada sobre el consumo de alcohol, sus trastornos y las discapacidades físicas y psiquiátricas asociadas.

Además, se incluyen cantidades y patrones de consumo, experiencias asociadas a la bebida, una clasificación de los trastornos por la ingesta de la misma o el historial familiar sobre problemas relacionados, entre otros.

También hay otras variables que no están directamente relacionadas con las anteriores y que también se utilizan para el análisis.

I.4 Estructura de la memoria

Para facilitar la lectura de esta memoria, se incluye a continuación, un breve resumen de cada capítulo:

-Capítulo 1: Aprendizaje máquina.

Introducción a los conceptos de aprendizaje máquina, clasificador Naïve Bayes y validación cruzada Leave-one-out.

[1]Encuesta Epidemiológica Nacional sobre Alcohol y Condiciones Relacionadas

-Capítulo 2: Procesado de datos.

Dedicado a la enumeración y explicación de los pasos seguidos durante el proyecto.

-Capítulo 3: Resultados.

Exposición y análisis de los resultados obtenidos.

-Capítulo 4: Conclusiones y líneas futuras.

Conclusiones a partir de los resultados obtenidos y líneas futuras de trabajo para mejorar los resultados.

-Capítulo 5: Presupuesto del proyecto.

-Apéndice A

Programas ejecutados para la realización del proyecto.

Capítulo 1

Aprendizaje máquina

1.1 Aprendizaje máquina

La definición de "Aprendizaje" puede ser tan simple como decir:

Aprendizaje = adquirir o aumentar el conocimiento.

Otra definición del mismo es:

El aprendizaje denota cambios en el sistema que permite que se realice la misma tarea más eficiente y eficazmente la próxima vez [Simon, 1983].

Y una más elaborada:

Un programa de ordenador aprende a partir de una experiencia E a realizar una tarea T (de acuerdo con una medida de rendimiento P), si su rendimiento al realizar T medido con P, mejora gracias a la experiencia E [Tom Mitchell, Machine Learning].

Si bien el aprendizaje es algo innato en los seres vivos, explicar las razones y el proceso del mismo para conseguir que una máquina "aprenda" es algo más complicado.

El aprendizaje máquina es una rama de la Inteligencia Artificial cuyo objetivo es desarrollar técnicas que permitan a un ordenador "aprender", o dicho de otra forma, generalizar comportamientos en base a una determinada información que se le suministra. Esto es, a partir de una serie de muestras y ejemplos que se tienen sobre cierta observación y mediante el uso de algoritmos para entrenar a la máquina, se debe obtener el mejor modelo que permita predecir el comportamiento ante el suceso en estudio.

El número de datos que se suelen manejar en estos casos es demasiado grande, lo cual hace imposible que las personas puedan realizar un análisis de éstos y obtener un resultado fiable.

Pero gracias a las técnicas de Aprendizaje automático, se pueden realizar análisis de grandes cantidades de datos y variables para poder así, extraer reglas o modelos válidos.

Existen varios tipos de aprendizaje en función del uso que se dé a los datos o de cómo se llegue al modelo que permita describir el caso que se está estudiando.

En este proyecto, el tipo de aprendizaje utilizado es el **"Aprendizaje Inductivo"**, en el que se obtienen modelos o leyes generales a partir de la generalización de ejemplos simples.

A continuación se definen algunos conceptos básicos:

-*Muestras:* son los ejemplos simples utilizados en el análisis. También pueden aparecer como: *ejemplos de entrenamiento, observaciones…*

El elemento que determina si una solución es o no viable, es la existencia de un conjunto de muestras adecuado y suficiente.

Normalmente se busca que dichas observaciones sean representativas del concepto buscado, por lo que deberán ser datos parecidos entre sí y que puedan ser descritos por un mismo conjunto de atributos.

-*Representación:* es un punto clave dentro del problema de aprendizaje. Suele ser de dos tipos, dependiendo de las condiciones y del método elegido: *representación simbólica* o *representación numérica*

Dentro de la Representación, es necesario definir los siguientes conceptos:

·Clases: en los problemas de representación binaria, las clases utilizadas suelen ser la clase positiva y la clase negativa.

·Atributos: definen las características de las muestras.

·Valores de los atributos: permiten describir a los atributos. Hay diversas formas de describir un atributo, y se puede realizar una definición más detallada o más general en función de las necesidades del problema.

Incluir o no ciertos atributos o ciertos valores de atributos, condiciona fuertemente el resultado final.

-*Etiquetado:* ya que los sistemas de aprendizaje pretenden mejorar su comportamiento a partir de la experiencia acumulada, el etiquetado ayuda a separar lo erróneo de lo acertado (hablando para el caso binario).

Siempre que se hable de un etiquetado previo de las muestras, se está hablando de **aprendizaje supervisado,** donde cada observación incluye un valor de la clase a la que corresponde.

En el caso contrario, está el aprendizaje no supervisado, que como se intuye, no tiene un etiquetado previo de muestras, o dicho de otra forma, el conjunto de observaciones no tiene clases asociadas, pero este tipo de aprendizaje no será utilizado en este proyecto.

-*Pre procesado:* en muchas ocasiones ocurre, que los ejemplos no están en un lenguaje apropiado para ser utilizados. En estos casos es necesario pasar de la representación inicial a la representación que utilizará el sistema de aprendizaje.

Entre otras cosas, el pre procesado se encarga de eliminar el ruido o los errores que pueda haber entre las muestras.

Como ya se ha introducido en la definición de pre procesado, los datos no se suelen usar tal y como se obtienen inicialmente y existen dificultades a la hora de manejarlos:

-puede existir ruido, esto es, ejemplos mal clasificados (dos ejemplos coinciden en sus valores de atributo pero poseen un valor de clase distinto y no hay forma de diferenciarlos)

-puede haber demasiadas variables, lo que a veces es contraproducente

-pueden haber datos que no aportan nada

-o se pueden encontrar datos incompletos, erróneos o inciertos

1.1.1 Aprendizaje inductivo supervisado

El aprendizaje supervisado es útil para la *Predicción,* se aprende un clasificador y sirve para explicar las causas que llevan a tomar una decisión.

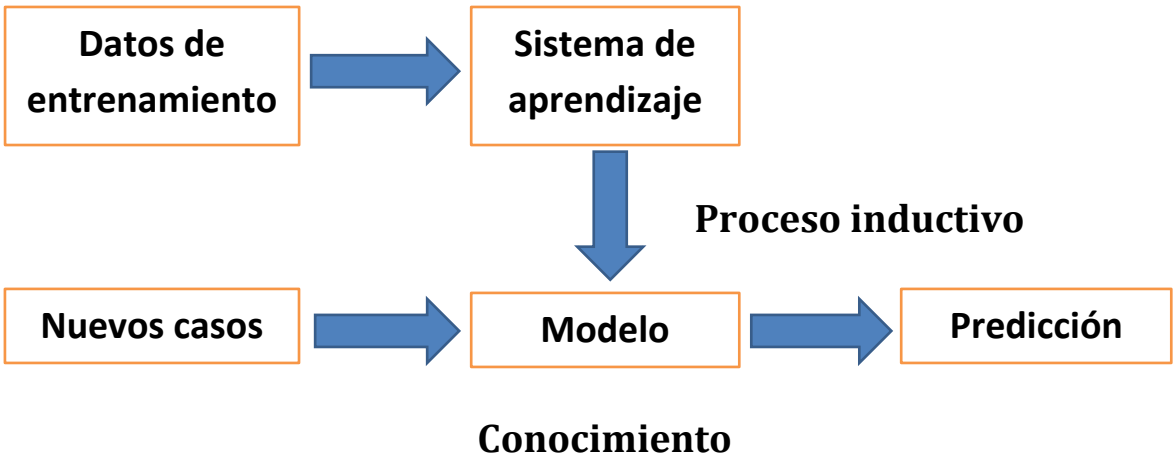

Figura 1.1: Aprendizaje Supervisado

Hay dos tipos de aprendizaje supervisado, atendiendo al número y tipo de clases:

-Clase discreta: conocida como clasificación. Dentro de este apartado, se encuentran varios tipos de clasificación, como la *Clasificación Binaria,* donde las clases posibles son +1 y -1; *Clasificación Multi-clase* cuya resolución, compleja en su origen, se realiza combinando Clasificadores Binarios, o la *Clasificación Multi-etiqueta,* que resuelve el problema que plantea la existencia de un mismo patrón con unos determinados atributos asociado a más de una clase.

-Clase continua: las salidas pueden tomar valores dentro de un conjunto infinito de posibilidades. Se puede hablar de regresión o regresión vectorial.

El tipo de clase utilizada en este proyecto es la clase discreta, por el tipo de datos que se van a manejar y la predicción (se intenta suicidar o no).

Como ya se ha comentado en el punto anterior, el aprendizaje supervisado trata de llegar a generalidades a partir de ejemplos simples, como se explica a continuación.

Se tiene una serie de muestras independientes e idénticamente distribuidas:

$$\{(x_1,y_1), (x_2,y_2), \dots , (x_i,y_i)\}$$

donde *x* son las variables e *y* la predicción a la que se quiere llegar.

El objetivo es predecir *y* dado un determinado valor de *x* pero, ¿cómo hacerlo?

Hay tres posibles modelos de resolución:

-Modelo p(x,y) (complicado)

-Modelo p(x|y)

-Obtener el mejor *y* para un *x* dado y llegar a una función del tipo $y=\delta(x)$, siendo $\delta(x)$ la respuesta producida por la máquina de aprendizaje.

Las dos últimas formas son parecidas, pero en adelante, se usará la tercera por ser la más sencilla de todas.

Pero a pesar de ser un método resolutivo relativamente sencillo, no se puede aplicar directamente sobre los datos de los que se dispone, ya que se plantean una serie de dificultades que han de resolverse.

Ya se ha dicho que debe obtenerse el mejor *y* dado un valor de *x*, pero ¿respecto a qué?. Llegados a este punto, es necesario incorporar un nuevo concepto, el de *Mínimo riesgo*.

Se debe aproximar δ(x) lo más fielmente posible a y y para ello se ha de buscar el valor óptimo a través de la ecuación:

$$\delta_{opt} = \underset{\delta}{arg\,min} \sum_y \int L(\delta(x), y) p(x, y) dx \tag{1.1}$$

siendo L la función de pérdidas. Para minimizar esta expresión, se necesita saber $L(\delta(x), y)$ y p(x,y), pero no se conoce la densidad de probabilidad conjunta y calcularla es muy complicado. La opción más factible es estimarla a partir del conjunto de muestras de las que se dispone, o lo que es lo mismo, estimar $\hat{p}$(x,y) y resolver la siguiente expresión:

$$\hat{\delta}_{opt} = \underset{\delta}{arg\,min} \sum_y \int L(\delta(x), y) \hat{p}(x, y) dx \tag{1.2}$$

De esta forma, se ha transformado un problema de clasificación, en un problema de estimación de densidad, pudiendo usarse las reglas del aprendizaje máquina clásico. Pero el problema no acaba aquí, ya que se tienen una serie de limitaciones al realizar esta operación. Al calcular la estimación de $\hat{p}$(x,y), se está asumiendo un riesgo que se debe cuantificar, y la forma en la que se va a hacer es mediante el uso de la definición de *Riesgo empírico*.

Se sabe que el riesgo asociado a una función δ es:

$$R(\delta) = \sum_y \int L(\delta(x), y) p(x, y) dx \tag{1.3}$$

Si se observa esta definición, la función elegida durante el aprendizaje, debe seleccionarse en base a:

1. Un conjunto de funciones de aproximación (F), que definirá el espacio de aproximación.
2. Un número limitado de ejemplos (ejemplos de entrenamiento).
3. Una función de pérdidas $L(\delta(\mathrm{x}), \mathrm{y})$ entre la respuesta del sistema y la respuesta de la máquina de aprendizaje.

Obsérvese ahora el siguiente razonamiento:

"Para obtener una buena generalización basta con seleccionar los parámetros de la función de aproximación que aseguren el número mínimo de errores sobre el conjunto de entrenamiento".

Siguiendo dicha observación, se puede sustituir la función del mínimo riesgo por la correspondiente a la del *mínimo riesgo empírico*:

$$R_{emp}^{n}(\delta) = \frac{1}{n}\sum_{i=0}^{n} L(\delta(\mathrm{x}_i), \mathrm{y}_i) \tag{1.4}$$

esta función es el promedio sobre los datos de entrenamiento, donde $\delta(\mathrm{x}_i)$ es la salida, L es la función de pérdidas y p(x,y) se ha sustituido por la siguiente expresión (la estimación):

$$\hat{p}(\mathrm{x},\mathrm{y}) = \frac{1}{n}\sum_{i=0}^{n} d(x - \mathrm{x}_i, y - \mathrm{y}_i) \tag{1.5}$$

siendo d la función delta de Dirac.

Pero ahora, ¿cómo elegir los datos de entrenamiento que se van a usar en las ecuaciones (1.4) y (1.5) y que van a proporcionar el mejor modelo?

Existen muchos métodos para clasificar los datos de entrenamiento. En el siguiente apartado, se introducirá el algoritmo de Naïve Bayes, donde se explicarán las ventajas que lo hacen apropiado para este proyecto.

1.2. Algoritmo de Naïve Bayes

1.2.1. Aprendizaje bayesiano

Dentro del Aprendizaje supervisado, una de las técnicas más relevantes y utilizadas es la del Aprendizaje bayesiano debido a dos razones principalmente:

1) Calcula probabilidades explícitas para hipótesis

2) Proporciona una perspectiva útil para comprender otros algoritmos de aprendizaje que no manipulan explícitamente probabilidades.

Éste tipo de aprendizaje tiene numerosas características que lo hacen adecuado para ser utilizado en tareas de clasificación ya que ofrece un análisis cualitativo de los atributos y de otros valores que intervienen en el problema. Pero además ofrece también un análisis **cuantitativo**, siendo ésta la gran aportación de los métodos bayesianos, y que consiste en dar una medida probabilística de la importancia de estas variables en el problema (probabilidad explícita).

Otras características del aprendizaje bayesiano son:

-Cada muestra de entrenamiento va a modificar la probabilidad de la hipótesis incrementándola o disminuyéndola, lo que significa que, mientras otros algoritmos eliminan completamente las hipótesis que no concuerdan con algún conjunto de muestras, el aprendizaje bayesiano no las desecha por completo, lo cual produce una disminución de la probabilidad estimada sobre dichas muestras.

-El conocimiento a priori se combina con datos observados para determinar la probabilidad final de la hipótesis.

-Proporciona resultados con probabilidades asociadas.

-Nuevos casos pueden ser clasificados mediante la combinación de predicciones de varias hipótesis.

-Son métodos robustos al posible ruido de entrenamiento y a ejemplos erróneos o incompletos.

Los inconvenientes pueden resumirse en los siguientes:

-pueden requerir el conocimiento inicial de muchas probabilidades

-alto coste computacional, dependiente del número de hipótesis

Como ya se explicó en el punto 1.1, el conjunto de datos será dividido en dos subconjuntos: el de entrenamiento y el de test que serán representados de la forma <atributo, valor>, y donde se buscará la hipótesis (o función de clasificación) que aproxime los ejemplos.

Gracias al Teorema de Bayes, en los problemas de aprendizaje máquina, se pueden estimar las probabilidades a posteriori de cualquier hipótesis y, para ello, se va a utilizar el algoritmo de Naïve Bayes.

1.2.2. Clasificador de Naïve Bayes

El clasificador de Naïve Bayes clasifica los ejemplos de acuerdo a la hipótesis más probable que lo describa.

Se trata de un algoritmo de clasificación no lineal, pero la principal característica por la que se ha elegido este algoritmo es por ser un algoritmo válido para variables discretas.

Sea *x* el ejemplo que se quiere clasificar representado por los siguientes valores:

$$<a_1, a_2, \dots, a_n>$$

Dicha función de clasificación vendrá dada por:

$$v_{MAP} = argmax_{v_j \in V} P(v_j | a_1, a_2, \dots, a_n) \quad (1.6)$$

Donde v_j es el valor de la función *f(x)* en el conjunto finito *V*.

Si a esta expresión se le aplica Bayes:

$$v_{MAP} = argmax_{v_j \in V} \frac{P(a_1, a_2, \ldots, a_n | v_j) P(v_j)}{P(a_1, a_2, \ldots, a_n)} = argmax_{v_j \in V} P(a_1, a_2, \ldots, a_n | v_j) P(v_j) \quad (1.7)$$

En el caso de $P(v_j)$, estimar sus diferentes valores es un procedimiento "sencillo", basta con observar la frecuencia con la que cada valor v_j aparece en los datos de entrenamiento. Sin embargo, estimar cada valor de $P(a_1, a_2, \ldots, a_n | v_j)$ es algo más complejo, ya que se requiere un gran conjunto de datos de entrenamiento. Es por ello que se recurre a la hipótesis de independencia condicional:

"la probabilidad de observar el conjunto $(a_1, a_2, \ldots, a_n)$ *para una categoría en concreto a la que pertenece, es el producto de las probabilidades de sus valores por separado":*

$$P(a_1, a_2, \ldots, a_n | v_j) = \prod_i P(a_i | v_j) \quad (1.8)$$

Si se sustituye esta hipótesis en la ecuación (1.7), se obtiene la aproximación al clasificador de Naïve Bayes:

$$v_{NB} = argmax_{v_j \in V} P(v_j) \prod_i P(a_i | v_j) \quad (1.9)$$

donde v_{NB} es el valor de la salida del clasificador y $P(a_i | v_j)$ se estima con la frecuencia de los datos observados.

Una vez obtenidos los datos de entrenamiento, éstos deben ser evaluados. Evaluar un clasificador implica medir la calidad del mismo. El método utilizado en este proyecto es el de *Validación cruzada* y más en concreto, el método de *Leave-one-out* que se pasará a explicar en el siguiente apartado.

1.3. Validación cruzada

Antes de explicar qué es la *Validación cruzada*, debe tenerse claro por qué es necesario evaluar un clasificador:

-hay que saber cuál es el mejor clasificador, cuál produce la mejor clasificación

-hay que estimar el error esperado, saber cómo es de fiable.

Es necesario probar el modelo bajo todas las posibles circunstancias, pero en general, no se tienen los suficientes datos de entrenamiento como para cubrir todos los casos, por lo que se hace necesario obtener un modelo capaz de generalizar a partir de los datos disponibles.

1.3.1. Leave-one-out

La validación cruzada dejando uno fuera (Leave-one-out cross validation, LOOCV), consiste en hacer que con cada iteración se tenga un solo dato como dato de prueba y el resto como datos de entrenamiento.

Esto es, se dividen los datos de entrenamiento en N subconjuntos. Para el primer conjunto, se separa el primer dato del grupo (dato de prueba), se entrena el modelo con las muestras restantes y se calcula el error empírico.

Para el segundo conjunto se separa el segundo dato, se entrena el modelo con las muestras restantes y se calcula el error empírico.

Este proceso se repite tantas veces como subconjuntos haya (en general, N). El número de iteraciones es igual al número de muestras del subconjunto y para cada iteración se calcula el error. El resultado final será el obtenido al hacer la media aritmética de todos los errores del subconjunto:

$$E = \frac{1}{N}\sum_{i=1}^{n} E_i \qquad (1.10)$$

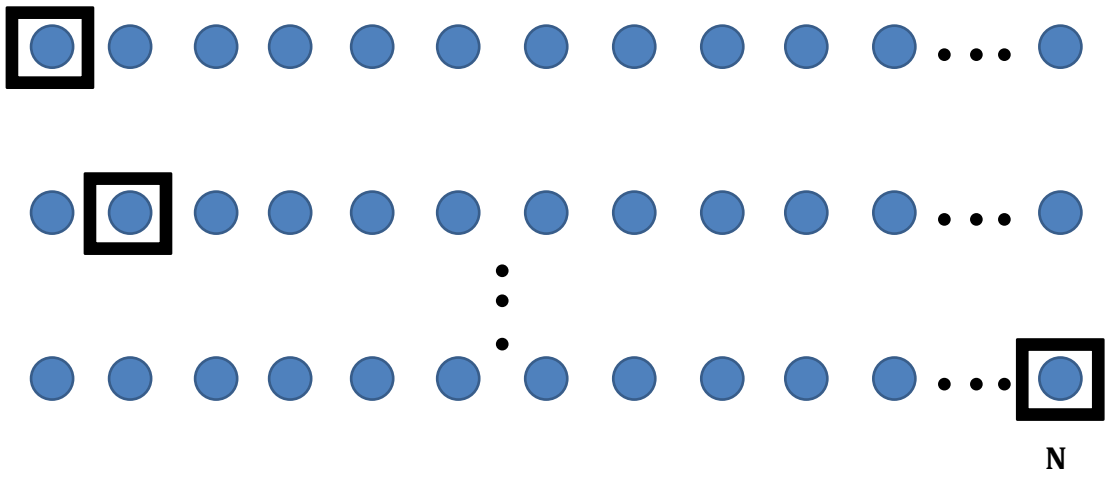

Figura 1.2: Algoritmo de Leave-one-out

Este método de validación utiliza todos los datos disponibles para entrenar, lo cual permite obtener un modelo que se acerque más a la realidad y con el que obtener unos datos más fiables.

Sin embargo, la carga computacional es bastante mayor, así como el tiempo requerido, ya que se realiza una iteración por cada muestra disponible.

A pesar de ello, la validación cruzada es la forma más adecuada de validar conjuntos de datos que no son especialmente grandes, permitiendo calcular tasas de errores pequeñas.

Una vez explicados los principales conceptos teóricos utilizados en el proyecto, en el siguiente capítulo se explicarán las etapas que lo componen, cómo y cuándo se aplican estos conceptos y los cálculos adicionales que se han de realizar para un correcto manejo de los datos, así como un análisis detallado de los resultados obtenidos.

Con ello, se llegará al decisor con menor probabilidad de error, pudiendo así predecir la probabilidad de intento de suicidio en un determinado paciente con la mayor exactitud posible.

Capítulo 2

Procesado de los datos

Una vez explicada la teoría sobre la que se asienta el proyecto, es el turno de explicar el trabajo realizado para obtener el resultado deseado, comenzando con el planteamiento del problema y siguiendo con el análisis de los pasos seguidos.

2.1 Planteamiento del problema

Como ya se explicó en el apartado I.2 de la Introducción, el objetivo de este proyecto es encontrar un clasificador que permita predecir si un sujeto va a intentar suicidarse o no. Para ello, se cuenta con una serie de datos obtenidos de las encuestas de NESARC, también explicado en el capítulo de Introducción (apartado I.3), y que ahora pasan a exponerse en detalle en el siguiente punto.

Comentar además, que el lenguaje de programación utilizado para este proyecto es MATLAB, un lenguaje de alto nivel.

2.1.1 Datos

Las encuestas NESARC han sido realizadas a un conjunto representativo de la población estadounidense, 43093 personas. De este conjunto, 29340 personas nunca han pasado por un período de depresión.

De las 13753 restantes, a la pregunta sobre intento de suicidio han respondido:

-1074 personas: SÍ

-12569 personas: NO

-83 personas: su respuesta es desconocida

Para cada sujeto se tienen 2991 entradas (preguntas contestadas) y hay 4 tipos de respuestas posibles:

-sí -no -desconocido -en blanco

En el siguiente apartado se explican los pasos seguidos hasta conseguir un clasificador con las características y propiedades deseadas.

2.2 Procesado de datos

Antes de intentar obtener cualquier tipo de clasificador, es necesario conocer los datos con los que realmente se cuenta, esto es, discernir entre los datos que serán de utilidad y los que sólo aportarán redundancia, provocando una disminución de la calidad del predictor. Además, es necesario separar los datos en dos grupos: por un lado los datos de *entrenamiento,* con los que la máquina será entrenada y por otro, los datos de *test* para probar si el predictor calculado es o no válido.

Todo esto puede resumirse en los dos siguientes puntos:

-hay gran número de variables con características de ruido que deben controlarse para no restar calidad al clasificador

-cualquier intento de construir un clasificador puede resultar fallido.

Por eso, para obtener un clasificador preciso es necesario:

-un algoritmo de selección de características: Selección hacia delante

-ordenar las características de antemano. Para ello se calculará la información mutua, que permitirá eliminar aquellas variables cuya información sea redundante

-un algoritmo simple de clasificación, que como ya se ha comentado en el capítulo anterior, es el de Naïve Bayes.

2.2.1 Pre procesado de los datos

La etapa de pre procesado comprende los siguientes pasos:

1. Los datos se encuentran almacenados en el archivo *"w1 codebook.pdf"*, el cual incluye las preguntas realizadas y las contestaciones (*Item value and description*), la frecuencia de cada respuesta (*Frecuency*) y la posición (*Source code*).
   Las respuestas cuyo valor sea desconocido (unknown) serán eliminadas, ya que su aporte no es positivo. La forma de eliminar estas variables consiste en asignar valores comprendidos entre 0 y 1 para las respuestas que contienen información útil, y -1 y -2 a las respuestas del tipo "unknown" y en blanco, respectivamente. (Véase *Preprocess.m* (Apéndice A.1))

2. Una vez se han "filtrado" los datos, es necesario dividirlos en dos grupos:
   -datos de entrenamiento: 10000 sujetos con los que entrenar a la máquina

-datos de validación: 3508 sujetos para probar el predictor calculado con los datos de entrenamiento.

Estos 13508 datos son el resultado de eliminar, sobre los 43093 iniciales, los sujetos que respondieron que nunca habían pasado por un período de depresión (29340), las respuestas dejadas en blanco y las respuestas cuyos valores son desconocidos, así como las anomalías que de ser tenidas en cuenta, restarían eficacia al predictor.

También en este paso se crea la variable **Y1: Intento de suicidio**, que es la variable a predecir.
(Véase *prepare_training1.m* (Apéndice A.2))

2.2.2 Procesado de los datos

Una vez concluido el pre procesado de los datos, se detallan a continuación los pasos seguidos para obtener el mejor predictor de intentos de suicidio.

A partir de los datos obtenidos en la etapa anterior, se generan los datos que serán utilizados para calcular el clasificador de Naïve Bayes (Véase *baseline1part0.m* (Apéndice A.4)).
Para ello se hace uso de las definiciones de *Información mutua* y *Entropía* que pasan a definirse a continuación.

Dentro del conjunto de datos que se maneja, hay unas variables que aportan más información a la hora de calcular el predictor que otras, por lo que se deben detectar y ordenar de forma que tengan prioridad sobre las que aportan menos información a la hora de seleccionar las variables que van a utilizarse para calcular el mejor predictor.
El cálculo de la información mutua entre dos variables aleatorias permite determinar la dependencia mutua entre dichas variables, o lo que es lo mismo, permite medir la de incertidumbre (entropía) de una variable X, conocida la

variable Y, por lo que todos aquellos resultados distintos de cero indicarán que la cantidad de información mutua que tienen con la salida es menor o mayor dependiendo de su valor.

$$I(X;Y) = \sum_x \sum_y p(x,y) log_2 \frac{p(x,y)}{p(x)p(y)} \geq 0 \ (bits) \quad (2.1)$$

Este proceso es válido para variables aleatorias discretas.

Aunque la forma más fácil de calcular la información mutua es a través de su relación con la entropía

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \quad (2.2)$$

Donde:

$$H(X) = -\sum_i p(x_i) log_2 p(x_i) \quad (2.3)$$

$$H(Y) = -\sum_i p(y_i) log_2 p(y_i) \quad (2.4)$$

$$H(X,Y) = -\sum_i p(x_i, y_i) log_2 p(x_{i,}, y_i) \quad (2.5)$$

(Véase entropy.m (Apéndice A.3))

Gracias al uso de estas definiciones, se podrán ordenar los datos en función de la información mutua de cada variable con la salida, esto es, poniendo en la primera posición la variable con mayor información mutua con la salida, en la segunda posición se coloca la segunda variable con mayor información mutua con la salida, y así sucesivamente con todas las variables de las que se dispone.

De esta forma, los grupos de variables que se utilicen, serán los que estén en las primeras posiciones (los de mayor información mutua con la salida), lo que aumentará las probabilidades de tener un error bajo al estar seleccionando las variables más adecuadas.

A partir de los datos ordenados en "baseline1part0.m", se quieren obtener distintos clasificadores mediante el algoritmo de Naïve Bayes que contengan 4 variables (Véase *baseline1part1.m* (Apéndice A.5)). Para ello, se coge un número determinado de variables cuya combinación dará lugar a varios clasificadores. Por ejemplo:

sea V el vector donde se almacenan las variables ordenadas según la información mutua con la salida

V1	V2	V3	V4	V5	V6	…	Vn

Si por ejemplo se cogen las 10 primeras variables, los clasificadores que se van a obtener serán los siguientes:

1º clasificador →	V1	V2	V3	V4
2º clasificador →	V1	V2	V3	V5
3º clasificador →	V1	V2	V3	V6
4º clasificador →	V1	V2	V3	V7
5º clasificador →	V1	V2	V3	V8
6º clasificador →	V1	V2	V3	V9
7º clasificador →	V1	V2	V3	V10
8º clasificador →	V1	V2	V5	V4

.
.
.

Figura 2.1: Ejemplo variables

A medida que el número de variables que se utiliza (N) aumenta, el tiempo de ejecución de baseline1part1.m aumenta y dependiendo de la capacidad de procesamiento del ordenador donde se ejecute el programa, puede llegar a tardar

incluso hasta días, por lo que en este proyecto se tomarán las primeras 10, 20 y 30 variables.

A continuación se muestra un ejemplo para explicar el proceso de selección de las variables que serán seleccionadas para formar parte de los clasificadores mediante el algoritmo de Naïve Bayes. En el ejemplo, se va a hacer uso de 2 variables en lugar de 4 para simplificar.

Ejemplo 1

Se han seleccionado las siguientes variables:

Variable 1: El paciente ha ido a consulta a causa de la depresión.
Variable 2: El paciente alguna vez ha robado.

Para estos dos supuestos presentes en la encuesta, las respuestas han sido las siguientes (obtenidas a partir del fichero *baseline1part0_1prueba_new*):

p(y = 'Sí'|x)

	En blanco	Desconocido	Sí	No
En blanco	0	0	18	47
Desconocido	0	3	2	1
Sí	0	7	90	194
No	0	2	144	275

Tabla 2.1: ***p(y = 'Sí'|Variable1)*** **y** ***p(y = 'Sí'|Variable2)***

p(y = 'No'|x)

	En blanco	Desconocido	Sí	No
En blanco	0	24	391	3099
Desconocido	0	9	2	2
Sí	0	5	58	277
No	0	26	884	4440

Tabla 2.2: ***p(y = 'No'|Variable1)*** **y** ***p(y = 'No'|Variable2)***

donde *y = Sí* indica que el sujeto ha intentado suicidarse e *y = No* indica que el sujeto no ha intentado suicidarse.

Si se calculan las frecuencias relativas para "Variable 1" y "Variable 2", se obtiene la siguiente tabla:

Variable 1	**Variable 2**	**p(y = 'Sí'|x)**	**p(y = 'No'|x)**
No	En blanco	1.49%	98.51%
Sí	No	4.40%	95.60%
No	No	5.83%	94.17%
No	Sí	14.01%	85.99%
Sí	No	41.19%	58.81%
Sí	Sí	60.81%	39.19%

Tabla 2.3: Frecuencias relativas

Los valores de *p(y = 'Sí'|x)* y *p(y = 'No'|x)* se han obtenido de la siguiente forma:

en el caso en el que *Variable 1= No* y *Variable 2=En blanco,* se tienen los valores 47 y 3099 respectivamente. Sumando ambos valores se obtiene 47 + 3099 = 3146.

Por último, si se calculan los porcentajes de 47 y 3099 sobre 3146 se llegan a los valores de 1.49% (47 x 100 / 3146 = 1.49) y 98.51% (3099 x 100 / 3146 = 98.51). Para el resto de los casos se ha seguido el mismo procedimiento.

Los valores de las frecuencias absolutas se obtienen en el conjunto de datos de entrenamiento. Para ello se observa el número de sujetos que han intentado suicidarse y los que no lo han intentado, obteniéndose los siguientes valores:

-Sujetos que han intentado suicidarse: 783

-Sujetos que no han intentado suicidarse: 9217

$$p(y = \text{'Sí'}) = 0.0783 \qquad p(y = \text{'No'}) = 0.9217$$

Una vez obtenidos los clasificadores de 4 variables, se eligen los 100 que menor error tengan (el hecho de elegir los 100 clasificadores con menor error y no los 50

o los 200 es indiferente, se ha elegido 100 para tener un número suficiente pero podría haberse elegido 120, 80… no hay un motivo concreto).

Los clasificadores son almacenados en una matriz de 7 columnas, siendo en la primera columna donde se almacena el error, en la segunda la probabilidad de No detección (probabilidad de tener un sujeto que puede suicidarse y no detectarlo), en la tercera la probabilidad de Falsa Alarma (probabilidad de detectar un sujeto como posible suicida y no serlo) y en las cuatro últimas columnas las variables a partir de las cuales se ha calculado el clasificador.

$Clasificador_1$	$Error_1$	ND_1	FA_1	V_{14}	V_{15}	V_{16}	V_{17}
$Clasificador_2$	$Error_2$	ND_2	FA_2	V_{24}	V_{25}	V_{26}	V_{27}
$Clasificador_3$	$Error_3$	ND_3	FA_3	V_{34}	V_{35}	V_{36}	V_{37}
				.			
				.			
				.			
$Clasificador_{100}$	$Error_{100}$	ND_{100}	FA_{100}	$V_{100\ 4}$	$V_{100\ 5}$	$V_{100\ 6}$	$V_{100\ 7}$

Figura 2.2: Clasificadores

Se define el error como

$$Error = ND \times factor + FA \tag{2.6}$$

donde *ND* es la probabilidad de No Detección y *FA* es la probabilidad de Falsa Alarma.

		Predicción	
		Intento	**No intento**
Verdad	**Intento**	OK	No detección
	No intento	Falsa alarma	OK

Figura 2.3: No Detección y Falsa Alarma

Cuanto menor sea el valor de ambas probabilidades, mejor será el decisor, pero existe un problema para conseguir esto, y es que si una de las probabilidades disminuye, la otra aumenta, por lo que habrá que llegar a un compromiso entre los dos valores, ya que aunque lo que más interesa es tener un valor de No detección lo más bajo posible, el de Falsa alarma tendría que aumentar y esto restaría calidad al decisor, por lo que tampoco sería bueno.

Factor es una variable usada para "penalizar" las no detecciones que pasa a explicarse brevemente en el siguiente párrafo.

El concepto de factor, surge para solucionar el problema de tratar cada caso de la misma forma, lo que conlleva a soluciones con modelos poco generales y de baja sensibilidad. Se trata de usar penalizaciones cuando se cometa un error:

-si es un falso positivo (se predice que el sujeto se intenta suicidar y no lo hace), la penalización es de 1 (factor = 1) ya que el hecho de predecir que un paciente se intenta suicidar y no lo haga no es un error grave

-si es un falso negativo (se predice que el sujeto no se intenta suicidar y sí lo hace), la penalización es el valor que se asigne a *factor* que en este proyecto variará entre 1 y 64, lo que dará lugar a distintos modelos.

A partir de los valores de No detección y de Falsa alarma se pueden calcular los valores de *Especificidad* (*E*) y *Sensibilidad (S)* como sigue:

$$S = \frac{Número\ de\ verdaderos\ positivos}{Número\ de\ verdaderos\ positivos + Número\ de\ falsos\ negativos\ x\ L} \qquad (2.7)$$

$$E = \frac{Número\ de\ verdaderos\ negativos}{Número\ de\ verdaderos\ negativos + Número\ de\ falsos\ positivos\ x\ L} \qquad (2.8)$$

Donde “número de falsos negativos” se refiere a la No detección y “número de falsos positivos” se refiere a la Falsa alarma.
Si se aplican estas definiciones al ejemplo anterior (Ejemplo 1) se obtienen los siguientes valores de Especificidad y Sensibilidad:

$$Sensiblidad = \frac{97}{783} = 0.124 \qquad Especificidad = \frac{9152}{9217} = 0.993$$

Resumiendo, llegados a este punto se tendrán varios conjuntos de clasificadores (matrices de 100x7, en el que cada fila es un clasificador), donde para cada conjunto se han tenido en cuenta las 10, 20 o 30 primeras variables con mayor información mutua con la salida (siendo la salida la variable de intento de suicidio) y se han elegido distintos factores de penalización (entre 1 y 64).

Después de agrupar los clasificadores para los distintos valores de N (número de variables utilizadas) (Véase *baseline1part2.m* (Apéndice A.6)), se va a proceder a mejorar los resultados hasta ahora obtenidos añadiendo más variables a los clasificadores, ya que parece absurdo utilizar simplemente 4 variables y no comprobar si la tendencia del error disminuye a medida que se le van añadiendo más. Para ello se hará uso del algoritmo de selección de características, Forward Selection, explicado brevemente en los siguientes puntos:

- ·se tiene un conjunto de n datos y d características
- ·se entrenan d clasificadores con una característica y se elige el mejor (la mejor característica)
- ·a continuación se entrenan d-1 clasificadores con dos características, la mejor del paso anterior y otra más
- ·se elige de nuevo el mejor clasificador, esto es, el mejor par de características
- ·se repite el proceso con tres características, cuatro, cinco… las necesarias hasta que el proceso comience a degradarse, o lo que es lo mismo, hasta un valor de error tolerable.

La razón por la que se ha utilizado este algoritmo en el desarrollo de este proyecto está motivada por el hecho de disponer de un amplio conjunto de variables para encontrar el clasificador de menor error ya que en caso contrario, el método adecuado habría sido *Background Selection.* Además, en comparación con otros, éste es un algoritmo más eficiente computacionalmente.

En el siguiente gráfico se muestra porqué se ha elegido un determinado número máximo de variables que se usarán para calcular los clasificadores aunque en realidad podría haberse elegido un número cualquiera. Para ello se va a ejecutar *baseline1part3.m* para distintos valores de "factor" y a analizar la tendencia del error a medida que el número de variables va aumentando.

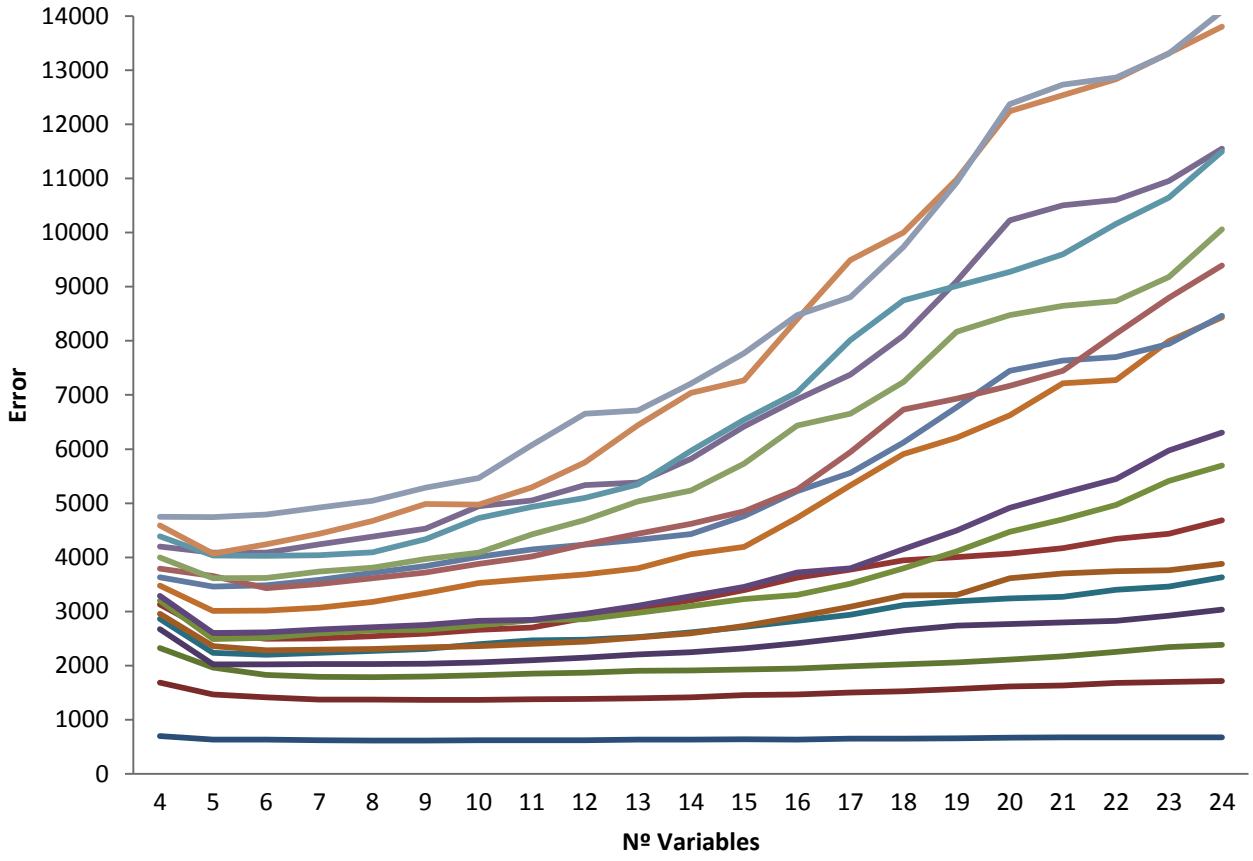

Figura 2.4: Gráfica de errores para cada nº de variables en función del valor "factor"

Cada línea se corresponde con el error de clasificación para un valor de factor dado y un número de variables comprendido entre 4 y 24.

Como puede apreciarse, independientemente del valor que se dé a "factor", a medida que el número de variables aumenta, el error tiene una tendencia creciente también, aunque no lo hace de forma totalmente lineal, ya que para algunos valores hay picos de bajada.

Lo que se busca con esto no es el valor mínimo del error, esto es, que para el caso de 5 variables, por ejemplo, se tenga el error más bajo de todos, sino que se busca un modelo que pudiendo llegar a contener hasta 14 variables, sea capaz de contener el mínimo error, aunque el promedio sea mayor.

Por tanto, la razón de tomar 14 variables como el valor máximo con el que se va a construir los clasificadores, responde al hecho de que entre los modelos posibles pueda estar el que tenga menor error. También puede ocurrir que este resultado se obtenga en modelos de 15 variables en adelante, pero como se ve en la gráfica, a medida que el número de variables se incrementa, es más complicado que suceda ya que el error aumenta demasiado y las probabilidades de encontrar el valor mínimo disminuyen (pero no es imposible).

Llegados a este punto, se van a ir añadiendo variables a los clasificadores que ya se tenían de 4 variables, siguiendo el algoritmo de *Forward Selection.* (Véase *baseline1part3.m* (Apéndice A.7)). El proceso es sencillo, como se explica a continuación.

Partimos de un grupo formado por clasificadores de 4 variables como el siguiente:

$Error_1$	ND_1	FA_1	V_1	V_2	V_3	V_4

Figura 2.5: Ejemplo de clasificador de 4 variables

al que se le van a ir añadiendo variables de forma progresiva. Pero entre todas las que se tiene, ¿cuáles se deben elegir?. La respuesta es sencilla. Como ya se explicó anteriormente, éstos han sido ordenados en función de la cantidad de información

mutua que tienen con la salida (Y1), por lo que se irán eligiendo grupos de variables de diferente tamaño (N1) que contengan las variables con mayor información mutua.

En el siguiente ejemplo se detalla el proceso para el caso en el que las variables son elegidas de un grupo de 100 (las 100 variables que tienen mayor información mutua con la salida).

Ejemplo 2

Partiendo de clasificadores como el de la Figura 2.5, se van a ir añadiendo variables, una a una, hasta llegar a 14 (ya se tienen 4, por lo que hay que añadir 10 más) elegidas de entre las 100 (N1=100) de mayor información mutua, como se explica en la siguiente figura:

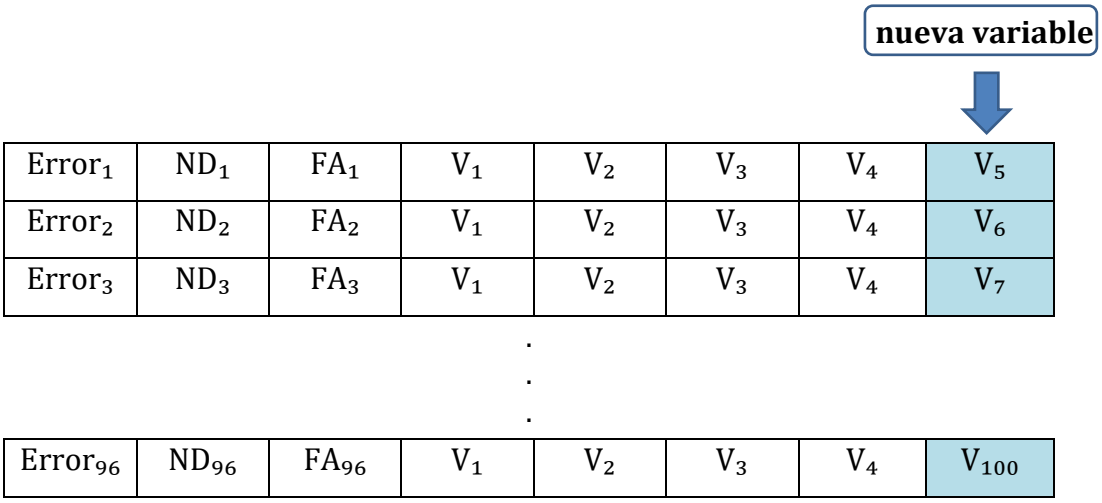

Figura 2.6: Clasificadores de 5 variables

Entre los 96 clasificadores obtenidos, se elige el que tenga un error menor (para este ejemplo, se supone que el clasificador de menor error es el que se obtiene al añadir la variable 7, representada en verde) y sobre ese, se añade otra variable más

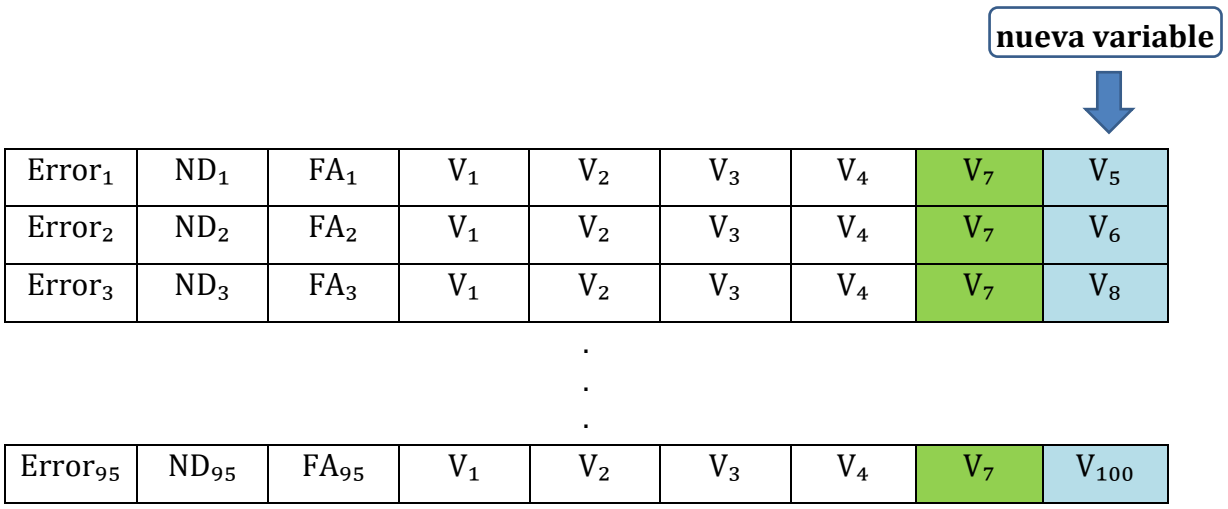

Figura 2.7: Clasificadores de 6 variables

Esta vez se han obtenido 95 clasificadores, se vuelve a elegir el de menor error y se añade otra variable más. Este proceso se repite hasta llegar a los 87 clasificadores de 14 variables.

El total de clasificadores que se obtienen para N1=100 son:

$$96_{(5\text{ variables})} + 95_{(6\text{ variables})} + 94_{(7\text{ variables})} + 93_{(8\text{ variables})} + 92_{(9\text{variables})} + 91_{(10\text{ variables})} + 90_{(11\text{ variables})} + 89_{(12\text{ variables})} + 88_{(13\text{ variables})} + 87_{(14\text{ variables})} = 915 \text{ clasificadores}$$

Cuanto mayor sea N1, más clasificadores se obtendrán, como puede observarse en la gráfica que se muestra a continuación.

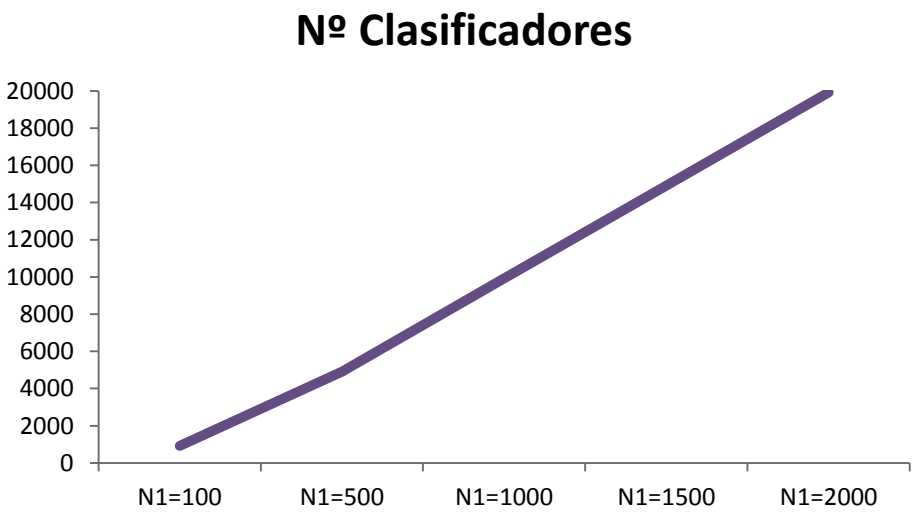

Figura 2.8: Nº de clasificadores en función del valor de N1

Pero estos valores se obtienen a partir de un solo clasificador. Como el lector recordará, anteriormente se ha explicado que se tiene un conjunto de 100 clasificadores a los que se le aplica el mismo proceso que acaba de describirse, por lo que para el ejemplo anterior, el total de los clasificadores será:

$$915 \times 100 = 91500$$

(915 clasificadores que se obtienen de aplicar el algoritmo de selección hacia delante a un clasificador, multiplicado por los 100 clasificadores que forman el grupo que se ha cogido). Pero si además se hace para valores de "factor" comprendidos entre 1 y 64, el número de clasificadores aumentaría hasta

$$91500 \times 64 = 5\,856\,000$$

El siguiente capítulo está dedicado al análisis de los resultados obtenidos tras seguir los pasos y ejecutar los programas que se acaban de explicar. Dependiendo de los valores seleccionados, se tendrá un determinado número de clasificadores entre lo que se elegirán los que mejor cumplan con los requisitos planteados para ser analizados.

Capítulo 3

Resultados

Durante este capítulo, se van a analizar los datos obtenidos tras ejecutar los programas explicados anteriormente, detallando las razones por las que se ha elegido unos determinados clasificadores y no otros.

3.1 Análisis de los resultados

En los siguientes apartados, se muestran las gráficas obtenidas para un valor determinado de la variable N1, así como los clasificadores seleccionados y sus variables asociadas. Dichas gráficas son curvas ROC, cuyo análisis proporciona los mejores modelos, así como los menos óptimos.

En cada curva, están representados los clasificadores para los siguientes valores de *factor*:

factor = (1, 2, 3, 4, 5, 8, 16, 32, 64)

3.1.1 Gráficas para N1 = 50, 100, 500 y 1000

Tras ejecutar *baseline1part3.m* para N1 = (50, 100, 500, 1000) y *sens_spec.m*, se obtienen las siguientes curvas ROC, donde los clasificadores situados por encima de la diagonal que divide a la gráfica, proporcionan mejores resultados que los situados debajo.

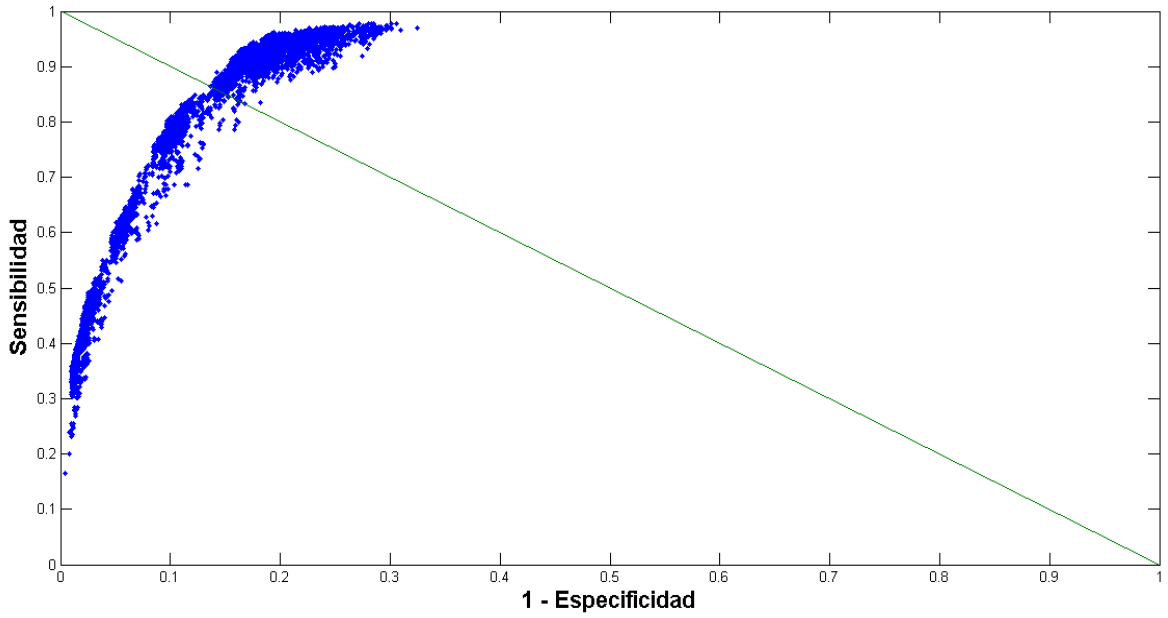

Figura 3.1: No Detección – Falsa Alarma para N1=50

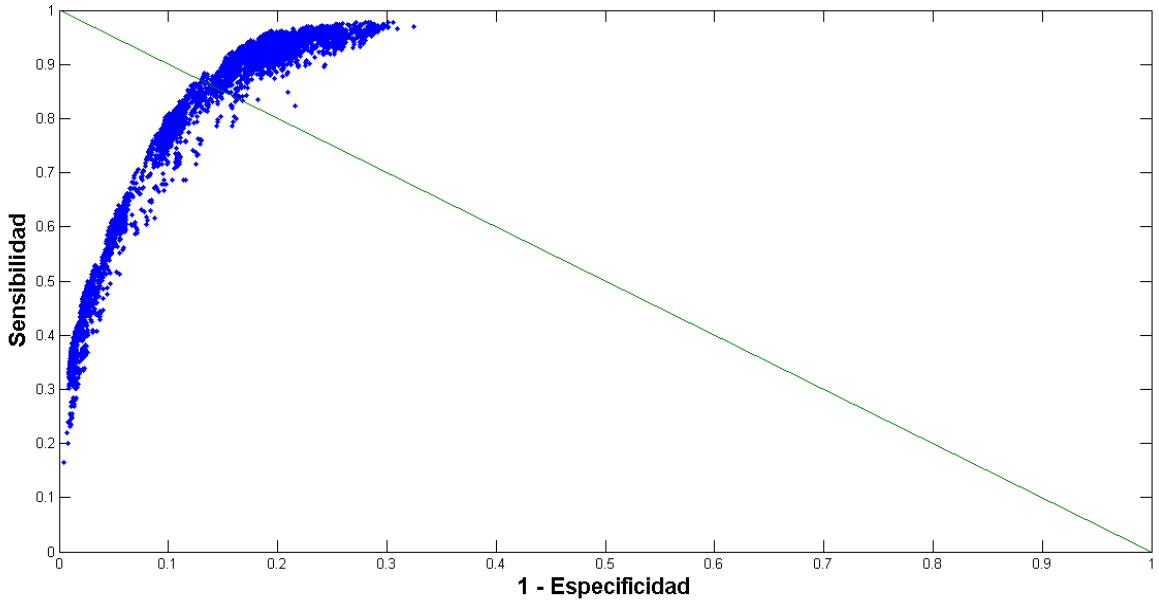

Figura 3.2: No Detección – Falsa Alarma para N1=100

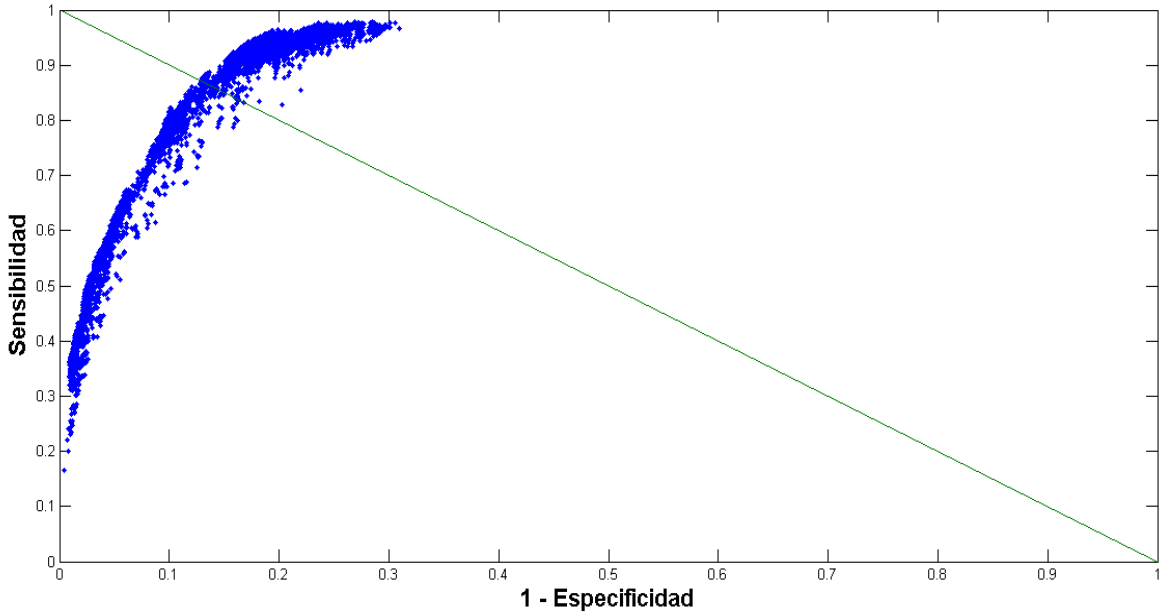

Figura 3.3: No Detección – Falsa Alarma para N1=500

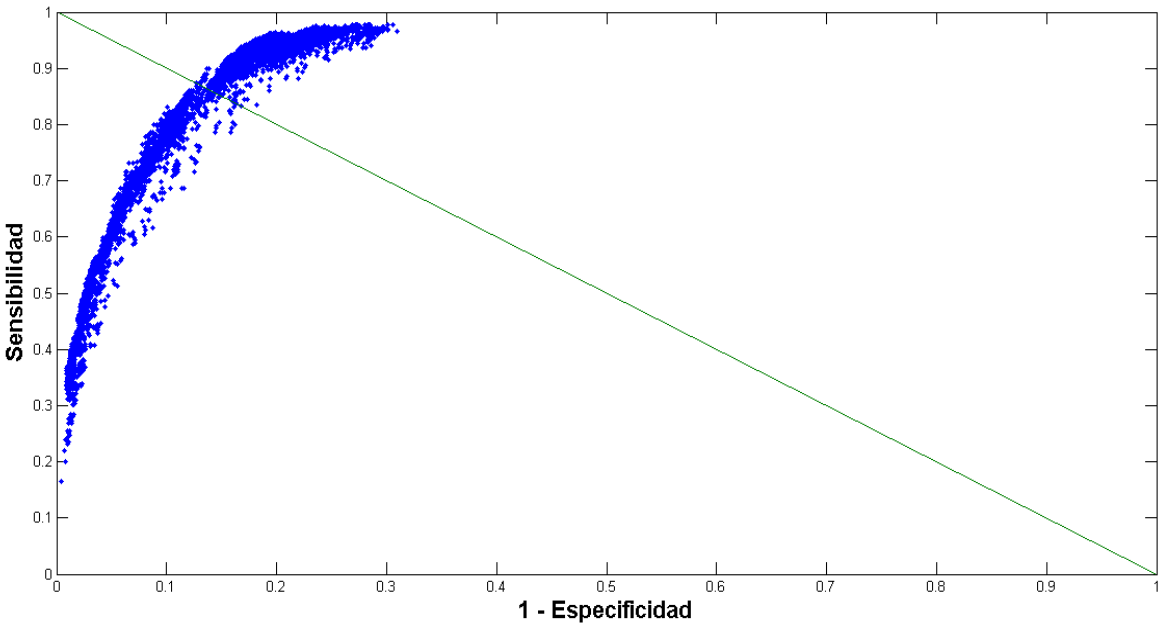

Figura 3.4: No Detección – Falsa Alarma para N1=1000

En los ejes de coordenadas se ha representado "Sensibilidad" (eje de ordenadas) frente a "1-Especificidad" (eje de abcisas), normalizados.

El clasificador ideal debería tener unos valores de No detección y Falsa alarma iguales a 0, o lo que es lo mismo, viendo la gráfica anterior, el clasificador ideal estaría situado en el punto (0,1), lo que representa un 100% de sensibilidad (sin falsos negativos) y 100% de especificidad (sin falsos positivos). Pero como ya se explicó en el apartado *2.2.2 Procesado de los datos* del capítulo anterior, eso es imposible. Por lo tanto, partiendo de este hecho, cuando se analicen las gráficas de resultados, se elegirán los clasificadores que se encuentren en las zonas más cercanas a estos valores.

Pero también hay que tener en cuenta, que predecir que un sujeto va a suicidarse y que resulte ser falso, no es un error tan grave como predecir que no se suicida y que finalmente lo haga. Es por ello que se intentará que la probabilidad de No detección sea lo más baja posible, pero sin olvidar que tampoco sería bueno tener un valor demasiado alto para la probabilidad de Falsa alarma.

Si se amplía la zona de interés, podrán distinguirse de forma más clara, las posiciones en las que se encuentran los clasificadores y hacer una selección más minuciosa de los mismos.

Para N1=50:

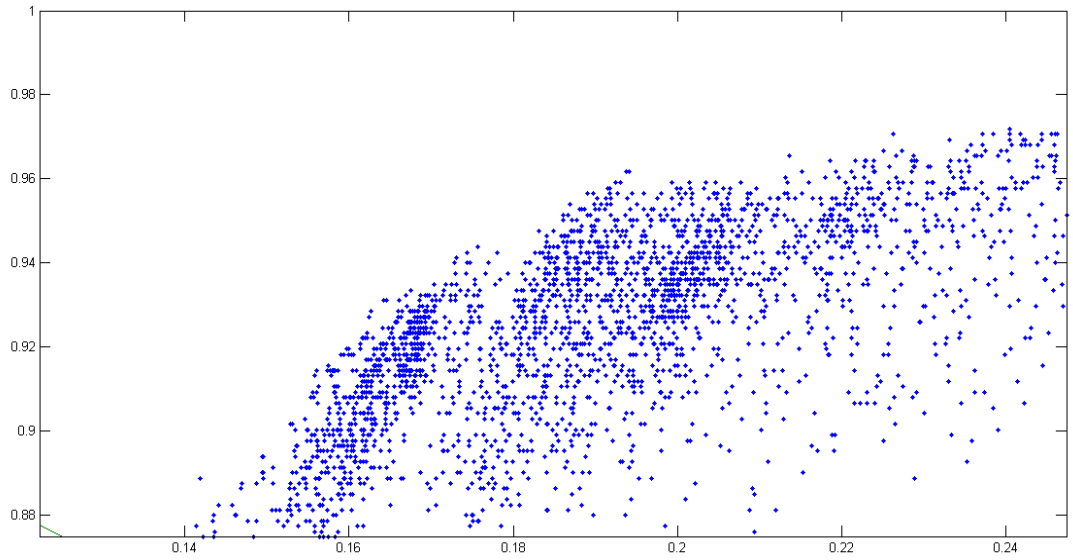

Figura 3.5: No Detección – Falsa Alarma para N1=50

Para N1=100:

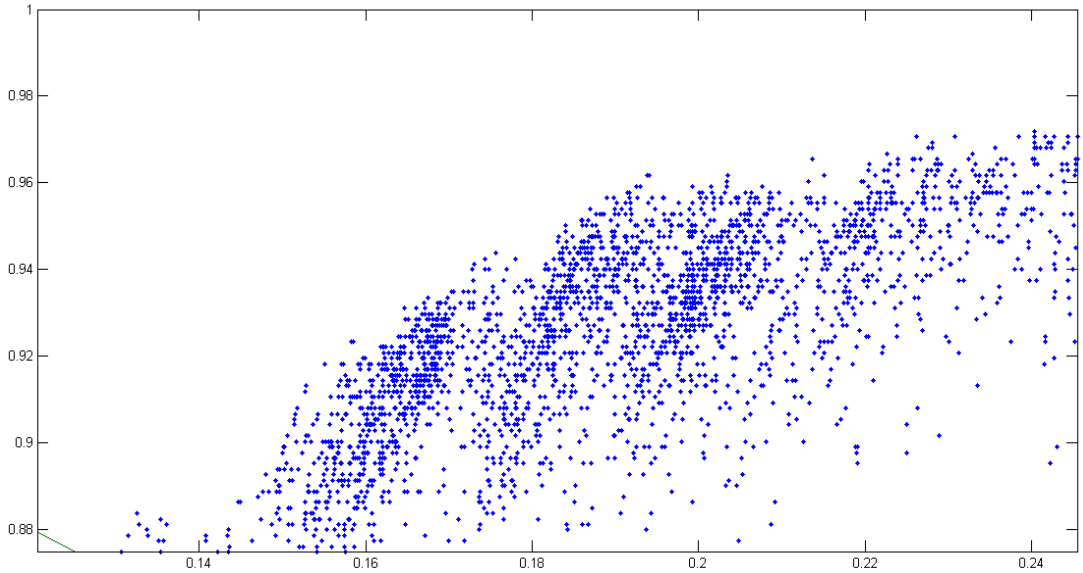

Figura 3.6: No Detección – Falsa Alarma para N1=100

Para N1=500:

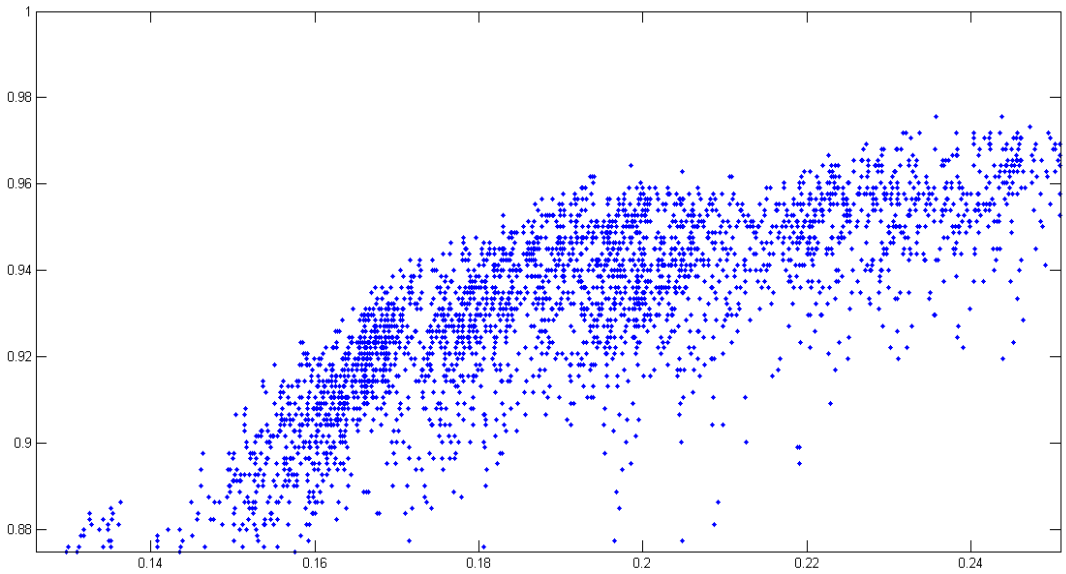

Figura 3.7: No Detección – Falsa Alarma para N1=500

Para N1=1000:

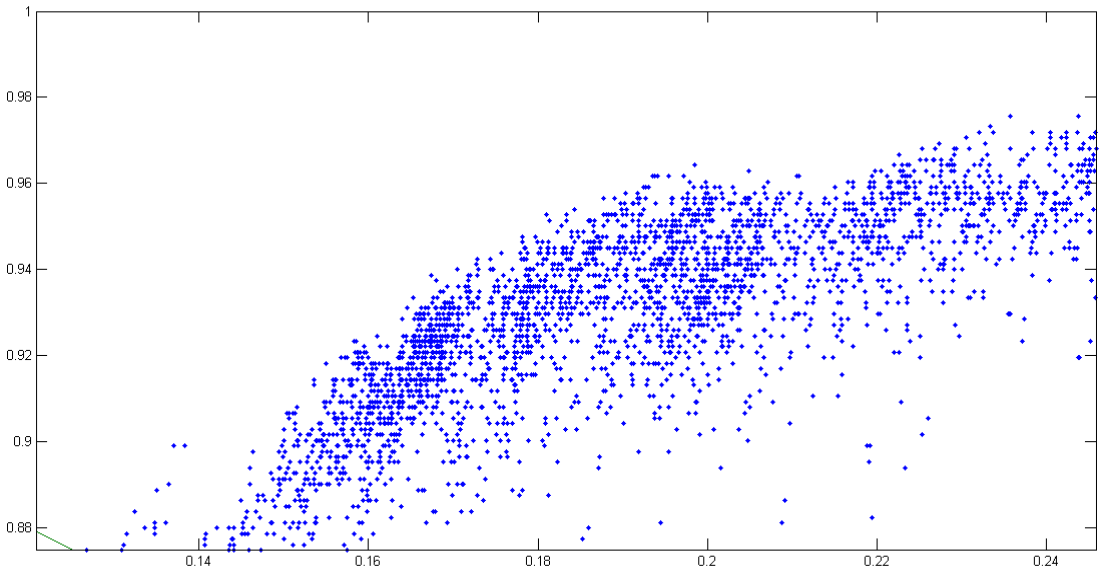

Figura 3.8: No Detección – Falsa Alarma para N1=1000

Para cada clasificador elegido, se van a mostrar los valores de Falsa Alarma, No Detección, Especificidad, Sensibilidad y el valor de Factor, y se analizará para qué cantidad de variables (N1) se obtiene el clasificador con los valores más próximos

a los deseados, esto es, con unos valores de No detección y Falsa Alarma lo más bajos posible, además de comprobar hasta qué punto influye que la variable "Factor" tenga un valor u otro.

Además, para cada clasificador, se van a obtener los modelos asociados a los mismos (tras ejecutar sens_spec.m), y en donde se van a poder observar qué variables son las que dan lugar a los mejores clasificadores.

3.1.2 Clasificadores y variables

A continuación se muestran los valores de los clasificadores seleccionados y las posiciones de las variables que los forman (en el apartado *2.2.2 Procesado de los datos*, se explicó que las variables son ordenadas en función de la información mutua que tengan con la salida, por lo que la variable situada en la posición 1 es la que más información mutua tiene con la salida, la 2 es la segunda variable que más información mutua tiene con la salida, y así sucesivamente).

Estas variables son las preguntas de la encuesta NESARC, las cuales tienen asociados unos determinados valores correspondientes a las respuestas de los sujetos que realizaron dicha encuesta y que también se van a exponer en este apartado para facilitar la comprensión del análisis por parte del lector.

N1=50

	Falsa Alarma	**No detección**	**Especificidad**	**Sensibilidad**	**Factor**
Clasificador1	0.194	0.038	0.806	0.962	16
Clasificador2	0.189	0.043	0.811	0.957	16
Clasificador3	0.214	0.034	0.786	0.966	64
Clasificador4	0.190	0.045	0.810	0.955	16

Tabla 3.1: Clasificadores para N1=50

los cuales, están asociados a las variables situadas en las siguientes posiciones:

	V1	V2	V3	V4	V5	V6
Clasificador1	1	2	3	4	28	
Clasificador2	1	2	3	5		
Clasificador3	1	3	4	6	2	16
Clasificador4	1	2	3	7	27	

Tabla 3.2: Variables para N1=50

N1=100

	Falsa Alarma	No detección	Especificidad	Sensibilidad	Factor
Clasificador1	0.168	0.066	0.832	0.934	8
Clasificador2	0.173	0.059	0.827	0.941	8
Clasificador3	0.226	0.029	0.774	0.971	64
Clasificador4	0.204	0.038	0.796	0.962	32

Tabla 3.3: Clasificadores para N1=100

y sus correspondientes variables:

	V1	V2	V3	V4	V5	V6
Clasificador1	1	2	4	5	21	28
Clasificador2	1	2	3	5	28	
Clasificador3	1	2	3	4	15	
Clasificador4	1	2	5	7	79	27

Tabla 3.4: Variables para N1=100

N1=500

	Falsa Alarma	No detección	Especificidad	Sensibilidad	Factor
Clasificador1	0.155	0.082	0.845	0.918	8
Clasificador2	0.188	0.043	0.812	0.957	32
Clasificador3	0.223	0.033	0.777	0.967	64
Clasificador4	0.178	0.052	0.822	0.948	16

Tabla 3.5: Clasificadores para N1=500

	V1	V2	V3	V4	V5	V6	V7	V8
Clasificador1	1	2	3	7	48	63	38	61
Clasificador2	1	2	3	5	120			
Clasificador3	1	2	4	6	160	3	44	123
Clasificador4	1	2	3	10	181	461		

Tabla 3.6: Variables para N1=500

N1=1000

	Falsa Alarma	No detección	Especificidad	Sensibilidad	Factor
Clasificador1	0.194	0.038	0.806	0.962	16
Clasificador2	0.181	0.047	0.819	0.953	16
Clasificador3	0.199	0.036	0.801	0.964	16
Clasificador4	0.165	0.066	0.835	0.934	8

Tabla 3.7: Clasificadores para N1=1000

	V1	V2	V3	V4	V5	V6	V7	V8
Clasificador1	1	2	3	4	234			
Clasificador2	1	2	3	6	652	803		
Clasificador3	1	2	3	4	234	165		
Clasificador4	1	2	3	6	413	553	181	335

Tabla 3.8: Variables para N1=1000

Llegados a este punto, se van a analizar los valores de los clasificadores seleccionados para poder entender el papel que desempeña el número de variables seleccionadas (N1) a la hora de obtener el clasificador más adecuado, así como la variable "factor" y las variables de la encuesta de las que se sirven los clasificadores de mejores características.

3.1.3 Análisis de probabilidades y variables

3.1.3.1 Variable Factor. Probabilidades de No detección y Falsa alarma.

Si se analizan los valores obtenidos en *factor* para cada N1, se observa que, en general (pueden existir clasificadores para los que esto no se cumpla, pero no es lo normal), cuanto mayor sea el valor de *factor*, el valor de No detección disminuye, llegándose a los valores más pequeños para un valor de factor de 64, el máximo utilizado en este proyecto.

Sin embargo, como ya se ha comentado con anterioridad, el hecho de que la probabilidad de No detección disminuya, implica que la probabilidad de Falsa alarma aumenta, por lo que lo óptimo no es elegir el valor más pequeño posible para No detección, de hecho, el valor o valores mínimos, no se encuentran en la zona próxima a los valores óptimos, sino que están situados en la parte derecha de la curva ROC. Por tanto, se debe llegar a un "acuerdo" donde el valor de Falsa

alarma no sea excesivamente alto, y esto se consigue en la mayor parte de los casos (independientemente del valor de N1), para unos valores de factor de 8 y 16, aunque como se muestra en las tablas, también se encuentran buenos clasificadores para valores de 32 y 64.

Los factores de valor menor a 8 no producen (salvo excepciones) buenos clasificadores, ya que éstos aparecen situados en la parte baja de la gráfica, y la gran mayoría por debajo de la diagonal, obteniéndose valores muy pequeños de Falsa alarma, pero en cuanto a No detección se refiere, los valores son inaceptables.

Por tanto, se puede decir respecto al factor, que su uso es del todo necesario y que los mejores valores y con los que se consiguen mejores clasificadores, son aquellos mayores o iguales a 8, ya que se obtienen unos valores de probabilidad de No detección bajos que se corresponden con unos valores de Falsa alarma aceptables.

En cuanto a las probabilidades de No detección y Falsa alarma, aunque no se encuentran grandes diferencias entre elegir un valor de N1 u otro, sí que se pueden observar algunos casos en los que habiendo obtenido un mismo valor de No detección, la probabilidad de Falsa alarma es menor en aquellos clasificadores donde N1 es mayor, como se muestra en la siguiente tabla:

	Falsa Alarma	**No detección**	**Especificidad**	**Sensibilidad**	**Factor**
N1 = 1000	0.194	0.038	0.806	0.962	16
N1 = 100	0.204	0.038	0.796	0.962	32
N1 = 1000	0.165	0.066	0.835	0.934	8
N1 = 100	0.168	0.066	0.832	0.934	8
N1 = 500	0.188	0.043	0.812	0.957	32
N1 = 50	0.189	0.043	0.811	0.957	16

Tabla 3.9: Probabilidades según N1

Además, en estos casos, el valor de la especificidad disminuye, cuando lo deseable, es que sea lo más grande posible, para estar más cerca del punto (0,1) de la curva ROC.

En este caso, sí se aprecia una mejora en aquellos clasificadores donde se han elegido un mayor número de variables (N1), ya que para un mismo valor de No detección, la probabilidad de Falsa alarma es menor y la especificidad mayor.

3.1.3.2 Variables de la encuesta

En la siguiente gráfica se muestran las posiciones de las variables (ordenadas según la información mutua con la salida) que componen los clasificadores seleccionados:

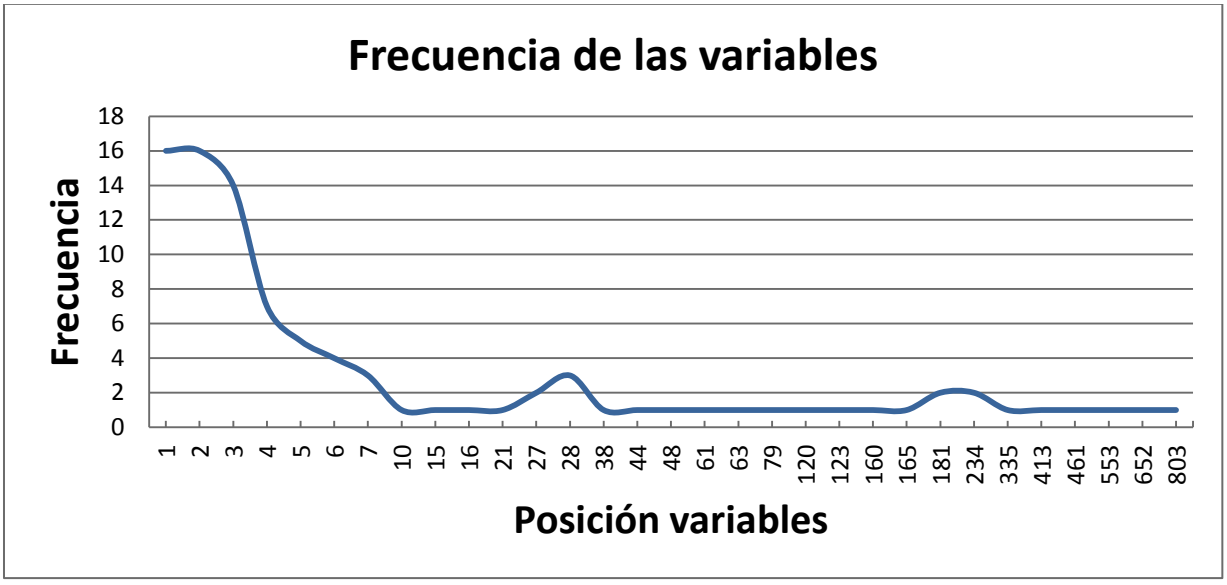

Figura 3.9: Frecuencia de las variables

donde se puede observar que las variables de mayor información mutua con la salida (1, 2 y 3) son utilizadas en todos o casi todos los clasificadores, y que en general, todas las variables que aparecen, se encuentran en posiciones "cercanas" a las variables con mayor información mutua (hay más de 3000 variables posibles, el 85.1% de las variables elegidas están en posiciones anteriores a la 80 y la más alejada está situada en la posición 803).

A continuación se muestran qué preguntas de la encuesta se corresponden con estas variables.

Posición de la variable	Pregunta de la encuesta	Respuesta	
1	Alguna vez ha pensado en suicidarse	-Sí -No -Desconocido -Nunca / Se desconoce si ha tenido un período de bajo estado de ánimo en el que no le importaría morir -NA	3566 10098 89 29340
2	Sentirse con ganas de morir	-Sí -No -Desconocido -Nunca / Se desconoce si ha tenido un período de bajo estado de ánimo en el que no le importaría morir -NA	4468 9163 122 29340
3	Ha estado una noche o más ingresado en el hospital por depresión	-Sí -No -Desconocido -El peor episodio no cumplió con los síntomas de depresión mayor -NA	1097 7695 25 34276
4	Ha acudido a urgencias por depresión	-Sí -No -Desconocido -El peor episodio no cumplió con los síntomas de depresión mayor -NA	906 7883 28 34276
5	Piensa a menudo sobre su propia muerte	-Sí -No -Desconocido -Nunca / Se desconoce si ha tenido un período de bajo estado de ánimo en el que no le importaría morir -NA	3487 10147 119 29340
6	Ha ido a un consejero/terapeuta/doctor/otra persona para ayudarle a mejorar su estado de ánimo	-Sí -No -Desconocido -El peor episodio no cumplió con los síntomas de depresión mayor –NA	4551 4243 23 34276
7	El médico le ha prescrito tratamiento para mejorar su estado de ánimo y que se sienta mejor	-Sí -No -Desconocido -El peor episodio no cumplió con los síntomas de depresión mayor –NA	3847 4939 31 34276

10	Ha acudido a urgencias por distimia	-Sí -No -Desconocido -El peor episodio no cumplió con los síntomas de distimia –NA	368 1813 24 40888
15	Alguna vez bebió alcohol para mejorar su estado de ánimo y sentirse mejor	-Sí -No -Desconocido -El peor episodio no cumplió con los síntomas de depresión mayor -NA	1697 7094 26 34276
16	Bebió alcohol en un período anterior a los últimos 12 meses para mejorar su estado de ánimo	-Sí -No -Desconocido -El peor episodio no cumplió con los síntomas de depresión mayor -NA	1557 7234 26 34276
21	Los episodios comienzan cuando experimenta efectos secundarios malos con la bebida	-Sí -No -Desconocido -El peor episodio no cumplió con los síntomas de depresión mayor –NA	383 8386 48 34276
27	Tuvo depresión en un período anterior a los últimos 12 meses	-Sí -No	7051 36042
28	Se ha auto medicado para mejorar su estado de ánimo en los últimos 12 meses	-Sí -No -Desconocido -El peor episodio no cumplió con los síntomas de depresión mayor –NA	122 8568 127 34276
38	El primer episodio ocurrió en los últimos 12 meses	-Sí -No -Desconocido -El peor episodio no cumplió con los síntomas de depresión mayor -NA	876 7919 22 34276
44	Ha sufrido depresión en un período anterior a los últimos 12 meses (sustancias inducidas descartadas)	-Sí -No	6967 36126
48	El médico relaciona los episodios anteriores a los últimos 12 meses con su enfermedad	-Sí -No -Desconocido -El peor episodio no cumplió con los síntomas de depresión mayor -No ha tenido episodios en período anterior a los últimos 12 meses -NA	843 7024 74 35152

61	Pasa más tiempo solo porque no quiere estar rodeado de otras personas	-Sí -No -Desconocido -El peor episodio no cumplió con los síntomas de distimia -NA	1678 512 15 40888
63	Se ha detenido en el pasado, meditó sobre el pasado	-Sí -No -Desconocido -El peor episodio no cumplió con los síntomas de distimia -NA	1580 605 20 40888
79	Se ha auto medicado en los últimos 12 meses para mejorar su estado de ánimo.	-Sí -No -Desconocido -El peor episodio no cumplió con los síntomas de distimia -NA	167 1989 49 40888
120	Tiene distimia crónica	-Sí -No	2189 40904
123	Ha sufrido depresión en un período anterior a los últimos 12 meses (enfermedades inducidas y sustancias inducidas descartadas)	-Sí -No	6373 36720
160	Trastorno maníaco antes de los últimos 12 meses (sustancias inducidas descartadas)	-Sí -No	1325 41768
165	Ha sufrido trastornos maníacos en períodos anteriores a los últimos 12 meses (enfermedades inducidas y sustancias inducidas descartadas)	-Sí -No	1236 41857
181	Sufre un trastorno de personalidad paranoica (crónico)	-Sí -No	2105 40988
234	Sufre trastorno de personalidad evasiva (crónico)	-Sí -No	995 42098
335	Ha sufrido depresión mayor en los últimos 12 meses (no jerárquico)	-Sí -No	3485 39608
413	Ha sufrido depresión mayor en los últimos 12 meses (enfermedad inducida descartada)	-Sí -No	3157 39936
461	Le cuesta ser abierto con otras personas y esto le causa enfado o problemas en el trabajo, o en la escuela, o con su familia, o con otras personas	-Sí -No -Desconocido	1230 40442 1421
553	Piensa que la gente que conoce es mejor que usted y esto le causa enfado o problemas en el trabajo, o en la escuela, o con su familia, o con otras personas	-Sí -No -Desconocido	492 41158 1443

652	Consume cannabis a menudo	-Sí -No -Desconocido	8172 34553 368
803	Bebió alcohol para evitar la ansiedad en los últimos 12 meses	-Sí -No -Desconocido -El peor episodio no cumplió con los síntomas de ansiedad generalizada -NA	173 2696 35 40189

Tabla 3.10: Encuesta

Observando las variables situadas en las primeras posiciones, se comprueba que, efectivamente, son las que guardan mayor relación con la variable de salida *Se ha intentado suicidar*:

"Alguna vez ha pensado en suicidarse" y "Siente ganas de morir"

A medida que se van alejando, el lector apreciará que no hay una relación tan directa de las variables que usan los clasificadores seleccionados con la variable de salida, pero que no dejan de ser hábitos, sentimientos o situaciones que pueden ser el origen de llegar a pensar, en un momento determinado, en suicidarse.

Pero en realidad, en este proyecto, no es tan importante conocer cuáles son las variables que desencadenan el intento de suicidio, como el hecho de conocer los datos numéricos que éstas conllevan, es decir, que si la primera variable es "Alguna vez ha pensado en suicidarse" o es "Consume cannabis a menudo" es indiferente y corresponde a otros profesionales (psiquíatras, psicólogos…) su estudio y análisis.

Por tanto, al igual que se hizo en el apartado del tema anterior *2.2.2 Procesado de los dato, Ejemplo 1,* se va a realizar una tabla para dos de los modelos obtenidos y analizar así, sus resultados con más detalle.

Modelo 1

(N1=1000, Clasificador 1)

-Falsa alarma: 0.194 -Especificidad: 0.806 -Factor: 16

-No detección: 0.038 -Sensibilidad: 0.962

-Variable 1: Alguna vez ha pensado en suicidarse

-Variable 2: Sentirse con ganas de morir

-Variable 3: Ha estado una noche o más ingresado en el hospital por depresión

-Variable 4: Ha acudido a urgencias por depresión

-Variable 5: Sufre trastorno de personalidad evasiva (crónico)

V1	V2	V3	V4	V5	Negativos	Positivos	P(y=No|x)	P(y=Sí|x)
Sí	Sí	Sí	Sí	Sí	72	174	0.293	0.707
No	No	No	No	No	114	2	0.983	0.017
Sí	No	No	No	Sí	217	5	0.977	0.023
Sí	No	Sí	No	Sí	7	3	0.7	0.3
Sí	No	Sí	Unk	Sí	1	0	1	0
Sí	No	Sí	Sí	Sí	10	5	0.67	0.33
No	Sí	Sí	Sí	Sí	35	5	0.875	0.125
No	No	Sí	No	Sí	80	2	0.976	0.024
No	No	No	No	Sí	2815	5	0.998	0.002
Sí	No	Sí	Sí	No	1	2	0.33	0.67
Sí	No	No	No	No	11	0	1	0
Sí	Sí	No	No	No	127	45	0.738	0.262
Sí	Sí	Sí	Unk	Sí	1	1	0.5	0.5
No	No	No	Sí	Sí	64	1	0.985	0.015
No	Sí	Sí	Sí	No	4	2	0.67	0.33

Tabla 3.11: Tabla Naïve Bayes - Modelo 1

Donde *Negativos* se corresponden con los que no han intentado suicidarse y han respondido a las 5 preguntas como se marca en la tabla y *Positivos* se corresponden con los que sí han intentado suicidarse y han respondido como se indica.

En esta tabla, se observan algunas combinaciones en las que tras ver las respuestas dadas por los sujetos que han realizado la encuesta, se puede determinar con un cierto grado de certeza, si se pueden llegar a suicidar o no.

Por ejemplo, si un paciente responde a las cinco preguntas que forman este modelo de forma afirmativa (primera combinación de la tabla Sí – Sí – Sí – Sí – Sí), la probabilidad de que se intente suicidar es del 70,7%, mientras que la probabilidad de que no intente suicidarse es del 29,3%.

Si por el contrario, el paciente responde a todas las preguntas de forma negativa (segunda combinación de la tabla No - No - No - No – No), la probabilidad de intento de suicidio baja hasta el 1,7% y la de no suicidio aumenta hasta el 98,3%.

El mismo razonamiento se puede hacer para el siguiente modelo seleccionado, y para todos en general. (*)

Modelo 2

(N1=1000, Clasificador 3)

-Falsa alarma: 0.199 -Especificidad: 0.801 -Factor: 16

-No detección: 0.036 -Sensibilidad: 0.964

-Variable 1: Alguna vez ha pensado en suicidarse

-Variable 2: Sentirse con ganas de mori

-Variable 3: Ha estado una noche o más ingresado en el hospital por depresión

-Variable 4: Ha acudido a urgencias por depresión

(*)El hecho de elegir las combinaciones que se muestran en la tabla no atiende a ninguna circunstancia en especial. Si se mostrasen todas las combinaciones posibles, habría que crear una tabla con un número de filas igual a **nºvariables^4**, esto es, para un modelo que tenga 5 variables: 5^4=625 filas. Para este proyecto se considera innecesario llegar hasta ese punto, ya que con lo expuesto el lector puede entender el procedimiento seguido.

-Variable 5: Sufre trastorno de personalidad evasiva (crónico)

-Variable 6: Ha sufrido trastornos maníacos en períodos anteriores a los últimos 12 meses (enfermedades inducidas y sustancias inducidas descartadas)

V1	V2	V3	V4	V5	V6	Negativos	Positivos	P(y=No|x)	P(y=Sí|x)
Sí	Sí	Sí	Sí	Sí	Sí	56	126	0.307	0.693
No	No	No	No	No	No	18	0	1	0
No	No	No	No	Sí	Sí	2679	5	0.998	0.002
Sí	Sí	No	Sí	Sí	No	8	6	0.57	0.43
Sí	No	Sí	Unk	Sí	Sí	1	0	1	0
No	No	No	No	No	Sí	96	2	0.98	0.02
No	Sí	Sí	Sí	Sí	Sí	30	4	0.88	0.12
No	Sí	Sí	Sí	Sí	No	5	1	0.83	0.17
No	No	No	Sí	Sí	Sí	57	1	0.983	0.017
Sí	No	Sí	Sí	No	No	1	2	0.33	0.67
Sí	No	No	No	No	No	2	0	1	0
Sí	Sí	No	Sí	Sí	Sí	23	13	0.64	0.36
Sí	Sí	Sí	Unk	Sí	Sí	1	1	0.5	0.5
No	No	No	Sí	Sí	No	7	0	1	0
No	Sí	No	Sí	Sí	Sí	20	3	0.87	0.13

Tabla 3.12: Tabla Naïve Bayes - Modelo 2

En el siguiente capítulo se valorarán los datos obtenidos y se expondrán las conclusiones a la vista de los resultados.

Capítulo 4

Conclusiones y líneas futuras

A lo largo de este proyecto, se ha realizado un estudio, en base a las preguntas respondidas por una parte de la población norteamericana, para determinar cuando un paciente puede intentar suicidarse o no.

Para ello, se han seguido una serie de conceptos y pasos, explicados en los capítulos 1 y 2 respectivamente, para obtener los resultados expuestos en el capítulo anterior con el fin de obtener, en función de una serie de parámetros, el mejor clasificador con el mínimo error posible.

La elección de unos parámetros sobre el resto de las opciones existentes, pasa a resumirse en este capítulo y antes de finalizarlo, se expondrán las líneas futuras de trabajo que podría seguir este proyecto.

4.1 Conclusiones

Tras realizar una serie de razonamientos para eliminar variables que no eran útiles, ordenarlas según la información mutua con la salida y aplicar algoritmos y conceptos, se ha llegado a una serie de clasificadores, de entre los que se han seleccionado algunos de los que mejor características poseen. Pero, ¿qué criterios han llevado a la elección de unos y no otros?

Como ya se explicó en el Capítulo 3, los programas con los que se obtienen los clasificadores (*baseline1part3.m* y *sens_spec.m*) se han ejecutado para unos valores determinados de N1 (grupo de variables seleccionado) y factor (penalización de las no detecciones):

-N1 = 50, 100, 500 y 1000

-factor = 1, 2, 3, 4, 5, 8, 16, 32 y 64

A la vista de las gráficas obtenidas para cada valor de N1 y factor, se pudo apreciar lo siguiente:

1) Para la variable N1, se ha observado que, a pesar de que las probabilidades de No detección y Falsa alarma son muy parecidas para cada uno de los cuatro valores estudiados y que incluso, las probabilidades de No detección son idénticas en algunos casos, hay variaciones en las probabilidades de Falsa alarma en cuanto a que para dos valores determinados de N1 y teniendo la misma probabilidad de No detección en ambos casos, se observa que la probabilidad de Falsa alarma es menor para el caso en el que N1 es mayor, acercándose más por ello, a los valores óptimos (véase tabla 3.9).

Por esto, a la hora de elegir entre un clasificador con un determinado número de variables y otro clasificador con un número mayor de variables, en principio, es preferible el segundo, porque teniendo la misma probabilidad de No detección, la

de Falsa alarma está más próxima a las deseada. Lo mismo ocurre con la Sensibilidad y Especificidad.

Pero no se puede hacer una generalización de esta conclusión.

Si se observa el Clasificador 1 de la Tabla 3.1 (N1=50), el Clasificador 4 de la Tabla 3.3 (N1=100) y el Clasificador 1 de la Tabla 3.7 (N1=1000), se puede observar, que para una misma probabilidad de No detección en los tres casos, si se compara la que se corresponde con N1=1000 con la de N1=100, efectivamente, la probabilidad de Falsa alarma ha descendido para el N1 mayor, pero si se compara con el obtenido para N1=50, se observa que la probabilidad de Falsa alarma es la misma que para N1=1000.

Por tanto, aunque parecen mejores aquellos clasificadores que han utilizado un número mayor de variables, y de hecho, en la mayor parte de los casos así es, el uso de clasificadores con menos variables no arrojan resultados significativamente peores, siendo incluso en algunos casos, como el que se acaba de ver, mejores.

2) En cuanto a la variable factor, a la vista de los resultados del capítulo anterior, se puede concluir con que los mejores clasificadores se obtienen para unos valores mayores o iguales a 5 (pueden existir excepciones, pero no es lo normal).

Esta afirmación queda del todo clara con la siguiente gráfica:

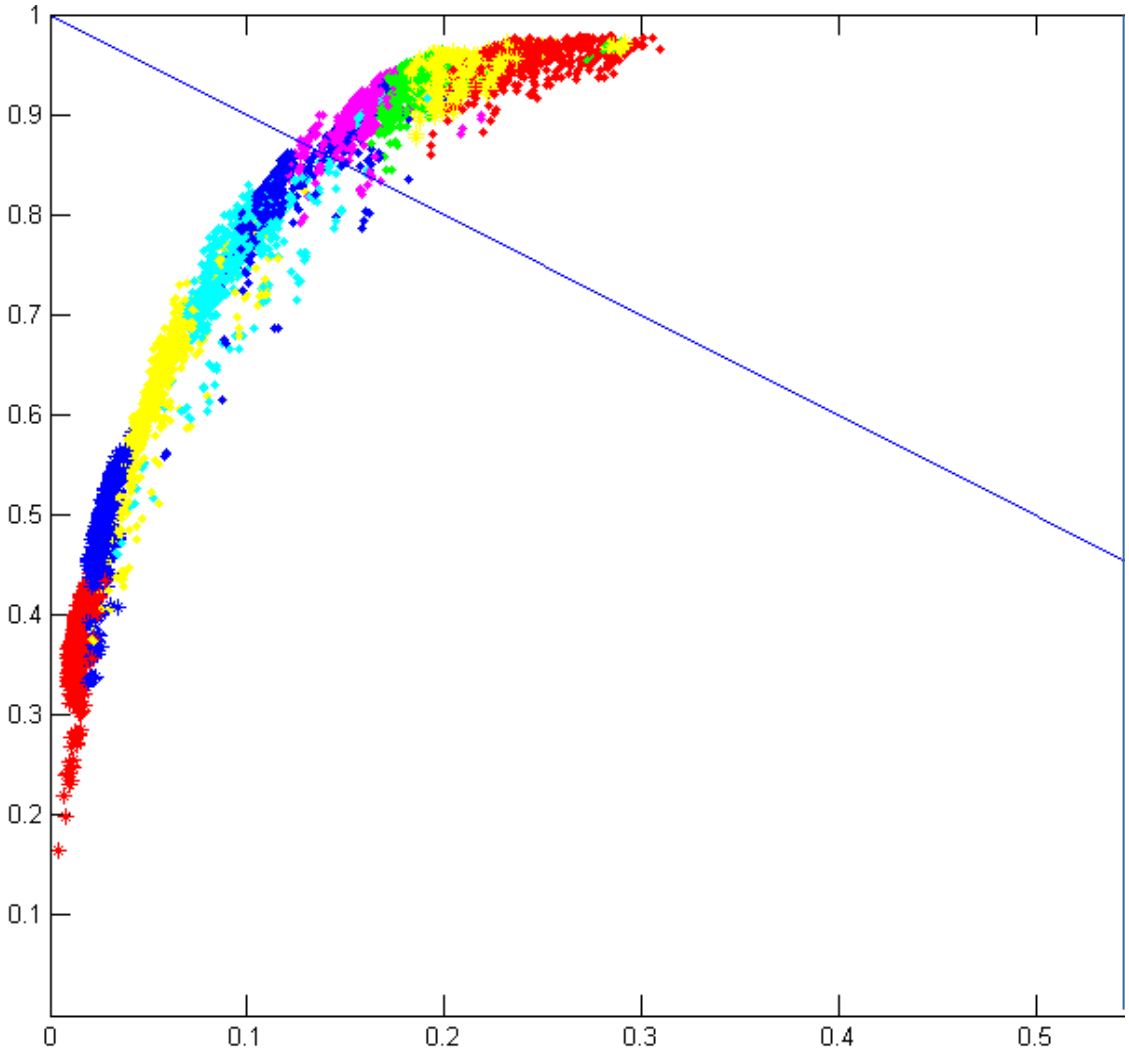

Figura 4.1: Distribución de la variable Factor

donde los factores están distribuidos de la siguiente forma:

factor = 1 factor = 3 factor = 8 factor = 32

factor = 2 factor = 4 factor = 16 factor 64

Como ya se comentó anteriormente, los valores óptimos estarían situados en el punto (0,1) de la gráfica, pero al no ser posible obtener clasificadores en este punto, se buscan los que estén en las zonas más cercanas. Como el lector podrá observar, en dichas zonas aparecen los factores de valores iguales a 5 y superiores, por lo que son éstos los valores que proporcionan los mejores clasificadores.

Por último, observando las variables que componen los modelos seleccionados, se puede comprobar, que éstos utilizan como mucho 8 variables por lo que, si se retrocede hasta el capítulo 2, apartado *2.2.2 Procesado de los datos*, parece que tiene sentido la explicación donde se decía que lo más "lógico" es buscar modelos que tengan un número mínimo de variables igual a cuatro, y que no sobrepase las catorce.

4.2 Líneas futuras de trabajo

Una de las líneas futuras de trabajo que pueden mejorar la propuesta presentada en este proyecto es la del uso de La Máquina de Vectores Soporte (en inglés Support Vector Machine o SVM), una técnica de aprendizaje que permite la clasificación de los datos de entrada en dos clases diferentes mediante un Hiperplano Óptimo de Separación.

Mediante el uso de esta técnica en su versión no lineal, se pretende alcanzar unos valores de Especificidad y Sensibilidad (y por tanto de No detección y Falsa alarma) más próximos al punto óptimo, como se indica en la siguiente figura:

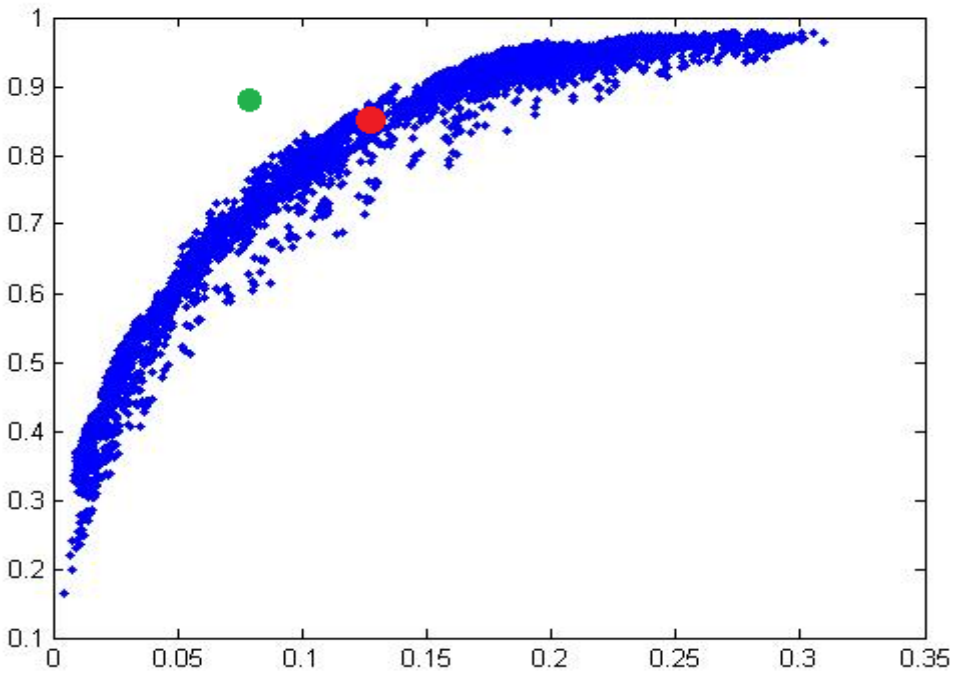

Figura 4.2: Modelo SVM

Donde el punto rojo representa la zona óptima de los clasificadores para el modelo con el que se ha trabajado durante el proyecto, y el punto verde representa la mejora que se podría conseguir aplicando el modelo SVM.

Por otro lado, observando la cantidad de variables de las que se disponen obtenidas a partir de la encuesta NESARC, y viendo las que realmente se han utilizado para crear modelos con tasas de error adecuadas, cabe preguntarse, hasta qué punto es necesario y conveniente el uso de más de 3000 variables, para llegar a un clasificador con las características deseadas.

Como se ha visto en el capítulo 3, los modelos de mejores características como mucho han necesitado de 8 variables, siendo las variables 1 y 2 utilizadas en todos ellos, la variable 3 se ha utilizado en el 90% de los casos, la variable 4 en casi el 50% y todas en general, se encuentran situadas entre las 800 variables con mayor información mutua.

Con estos datos, uno puede pensar que utilizando las variables más significativas en la detección del suicidio, como son las variables 1, 2, 3 y 4 y viendo que la tendencia para la elección del resto de variables no sobrepasa las 1000 primeras, sería lógico desechar todas las demás y trabajar en la creación de modelos más precisos a partir de las variables que realmente aportan información y obviar aquellas cuyo aporte es nulo, trabajando así con un grupo más reducido de variables y simplificando el problema.

Capítulo 5

Presupuesto del proyecto

En este capítulo, se presenta el presupuesto global para la realización de este proyecto, tanto gastos en material como en personal.

Se muestran a continuación las fases de las que se compone el trabajo:

Fase 1	Documentación	80 horas
Fase 2	Desarrollo del modelo	400 horas
Fase 3	Obtención y análisis de los resultados	200 horas
Fase 4	Redacción de la memoria	180 horas

Tabla 5.1: Fases del proyecto

A continuación se detallan los gastos en materiales: equipo informático y otros gastos que se puedan ocasionar.

Equipo informático	700 €
Otros gastos	400 €
TOTAL	1100 €

Tabla 5.2: Costes materiales

En función de las horas calculadas en la Tabla 5.1, se calcula el gasto personal como sigue:

Personal	Horas de trabajo	€/hora	Total
1 Ingeniero Técnico	840	20	16800 €

Tabla 5.3: Coste personal

Presupuesto total:

Concepto	Importe (€)
Coste personal	16800
Costes materiales	1100
Base imponible	17900
I.V.A (21%)	3759
TOTAL	21659

Tabla 5.4: Presupuesto total

El presupuesto total del proyecto asciende a la cantidad de 21659 €.

La ingeniera proyectista,

Fdo. Tamara Bouza López

Apéndice A

A continuación se muestran los programas en lenguaje Matlab utilizados en este proyecto.

A.1 preprocess.m

A través del programa *Preprocess.m* se genera una matriz, *M.dat*, donde se almacenan todos los modelos (parte comentada). Esta parte del programa se ejecuta una sola vez (a no ser que los datos de entrada con los que se cuentan cambien). A continuación, se eliminan las anomalías y las muestras donde la respuesta es desconocida.

Las variables se han pre procesado entre 0 y 1 para las respuestas que aportan información y las respuestas donde el valor es desconocido o dejadas "en blanco", se han codificado como -1 y -2, respectivamente. Los datos se almacenan en *data1.mat*.

```
clear
close all
format compact
load Posiciones
```

```
%La parte comentada, sólo se ejecuta una vez
% M=-2*ones(13753,2991);
% fid=fopen('Originalfiles/w1nesarcdata.txt', 'r');
% k2=1;
% for j=1:43093
%    k1=1;
%    if(rem(j,1000)==0)
%       [j k2]
%    end
%    c=fread(fid,3684,'uint8=>char');
%    if(c(2476)=='1' || c(2476)=='2' || c(2476)=='9' )
%       for i=1:length(P)-1
%          if(c(P(i,1))=='.')
%             k1=k1+1;
%          else
%             M(k2,k1)=str2num(c(P(i,1):P(i,2))');
%             k1=k1+1;
%          end
%       end
%       k2=k2+1;
%    end
% end
% fclose(fid);

load M

I0=[1898:1901];
Out0=[];
I1=[190 217 218 223 697 712 713 942 946 948 1762 1763 1764 1765 1766 2772 2831 2833];
Out1=[];
I2=[240 241 362 367 478 482 600 944 945];
Out2=[];
I3=[1913 2063 2136 2226 2317 2406];
Out3=[];
for i=1:2991
   if(find(I0==i))
      Out0=[Out0 find(M(:,i)==9)'];
   end
```

```
    %Remove the clear outliers
    if(find(I1==i))
        Out1=[Out1 find(M(:,i)==(max(unique(M(:,i)))-1))'];
%valores del vector sin repetir
    end
    %Remove the highly likely outliers
    if(find(I2==i))
        Out2=[Out2 find(M(:,i)==(max(unique(M(:,i)))-1))'];
    end
    %Encuentra posibles anomalías
    if(find(I3==i))
        Out3=[Out3 find(M(:,i)==(max(unique(M(:,i)))-1))'];
    end
end
Out0=unique(Out0);
Out1=unique(setdiff(Out1,Out0));
Out2=unique(setdiff(Out2,[Out0 Out1]));
Out3=unique(setdiff(Out3,[Out0 Out1 Out2]));
In=setdiff(1:13753,[Out0 Out1 Out2]);
M_in=M(In,:);
M_out=M([Out0 Out1 Out2],:);

v=zeros(4,2991);
for i=1:2991
    aux=unique(M_in(:,i));
    if(unique(aux-round(aux))==0)
        v(2,i)=1;
    end
    v(3,i)=max(aux);

    %Transformar las variables desconocidas en -1
    if(i==9)
        %nada
    elseif(i==12)
        %nada
    elseif(i==31)
        M_in(find(M_in(:,i)==98),i)=59;
        M_in(find(M_in(:,i)==99),i)=-1;
        M_out(find(M_out(:,i)==98),i)=59;
```

```
    M_out(find(M_out(:,i)==99),i)=-1;
  elseif(i==1915 || i==1918)
    M_in(find(M_in(:,i)==9997),i)=3;
    M_in(find(M_in(:,i)==9998),i)=52;
    M_in(find(M_in(:,i)==9999),i)=-1;
    M_out(find(M_out(:,i)==9997),i)=3;
    M_out(find(M_out(:,i)==9998),i)=52;
    M_out(find(M_out(:,i)==9999),i)=-1;
  else
    if(v(3,i)==9 || v(3,i)==99 || v(3,i)==999 || v(3,i)==9999 || v(3,i)==99999 || v(3,i)==999999 ||
v(3,i)==9999999)
      M_in(find(M_in(:,i)==v(3,i)),i)=-1;
      M_out(find(M_out(:,i)==v(3,i)),i)=-1;
    end
  end
  aux=unique(M_in(M_in(:,i)>=0,i));
  v(1,i)=length(aux);

  if(i~=1563)
    v(3,i)=max(aux);
    v(4,i)=min(aux);
  end

  %Subtracts the minimum value, so we start in 0 for the discrete random varaibles.
  if(v(4,i)>0 && v(2,i)==1)
    M_in(M_in(:,i)>0,i)=M_in(M_in(:,i)>0,i)-v(4,i);
    M_out(M_out(:,i)>0,i)=M_out(M_out(:,i)>0,i)-v(4,i);
    v(3,i)=v(3,i)-v(4,i);
  end

  %Si se divide entre el máximo, las variables están entre 0 y 1
  if(v(2,i)==1 && v(1,i)>1)
    M_in(M_in(:,i)>=0,i)=M_in(M_in(:,i)>=0,i)/v(3,i);
    M_out(M_out(:,i)>=0,i)=M_out(M_out(:,i)>=0,i)/v(3,i);
    v(3,i)=1;
  end

  %Preprocesado de las 16 variables continuas, entre 0 y 1
  if(v(2,i)==0)
```

```
    M_in(M_in(:,i)>0,i)=M_in(M_in(:,i)>0,i)-v(4,i);
    M_out(M_out(:,i)>0,i)=M_out(M_out(:,i)>0,i)-v(4,i);
    v(3,i)=v(3,i)-v(4,i);
    M_in(M_in(:,i)>=0,i)=M_in(M_in(:,i)>=0,i)/v(3,i);
    M_out(M_out(:,i)>=0,i)=M_out(M_out(:,i)>=0,i)/v(3,i);
    v(3,i)=1;
  end
end

save data1 M M_in M_out I0 I1 I2 I3 Out0 Out1 Out2 Out3 v P

format loose    %control formato de salida en display
```

A.2 prepare_training1.m

Haciendo uso de los datos almacenados tras la ejecución de Preprocess.m, es necesario dividirlos en dos grupos: por un lado, el conjunto de entrenamiento y por otro el conjunto de validación. Para ello se ejecuta *Prepare_training1.m*.

Además, se eliminan las variables que no tienen información sobre los pacientes y se crea Y1 (Intento de suicidio), la variable que se quiere predecir. Los datos son almacenados en *data_train1.mat*.

Este programa tampoco es necesario volver a ejecutarlo, ya que se quiere tener el mismo conjunto de test y de entrenamiento. Si la etapa de pre procesado cambiase de alguna forma, entonces sí se debería ejecutar de nuevo.

A.3 entropy.m

```
clear
load data1

%entropía conjunta de 2991 variables, información mutua entre las variables 1898(2476),
%1899(2477), 1900(2478), 1901(2479) y las otras 2991 variables
%crear vectores para almacenar los resultados
n=size(M_in,1);
T=zeros(1,2991);
H=zeros(1,2991);
```

```
for i=1:2991

  aux=unique(M_in(:,i));
  T(i)=length(aux);
  v=zeros(1,T(i));

  for j=1:T(i)

    v(j)=length(find(M_in(:,i)==aux(j)))/n;
  end

  H(i)=-sum(v.*log2(v));   %entropía

end

Hc=zeros(4,2991);

for k=1898:1901   %variables
  aux1=unique(M_in(:,k));

  for i=1:2991

    aux=unique(M_in(:,i));
    v=zeros(1,2*T(i));

    for j=1:T(i)

      v(j)=length(find(M_in(:,i)==aux(j) & M_in(:,k)==aux1(1)))/n;
    v(j+T(i))=length(find(M_in(:,i)==aux(j)&M_in(:,k)==aux1(2)))/n;

    end

    %k-1897---->valores de 1 a 4
    Hc(k-1897,i)=-sum(v(v>0).*log2(v(v>0)));
        end
end

I=ones(4,1)*H+H(1898:1901)'*ones(1,2991)-Hc;
%almacenar resultados
save entropy H Hc I T
```

A.4 baseline1part0.m

A partir de los archivos *Posiciones*, *data_train1.m* y *entropy*, genera el conjunto de datos que se usarán para el clasificador de Naïve Bayes.

```
function baseline1part0(out)
%out es la salida que se está calculando, su valor es 1 y se corresponde con la variable de Intento de
%suicidio

load entropy
load data_train1
load Posiciones

Variables_to_eliminate=[];
%De momento no se van a eliminar las variables, %Variables_to_eliminate[] se deja vacío

V2E=zeros(size(Variables_to_eliminate));
for i=1:length(Variables_to_eliminate)
  V2E(i)=find(P(:,1)==Variables_to_eliminate(i));
end

f=[0 ones(1,3) zeros(1,3) 4*ones(1,1898-7-1) 0 5*ones(1,2991-1898)];
[val,pos]=sort(I(out,:));

NN=13;
Q=[];
Q1=[];
Q2=[];
k=2991;

while(k>0)
%mientras haya características
  if(f(pos(k))==0)
    k=k-1;
  else
    if(T(pos(k))>NN)
      k=k-1;
    else
      if(length(find(pos(k)==V2E))==0)
```

```
            Q=[Q pos(k)-f(pos(k))];
            Q1=[Q1 pos(k)];
            Q2=[Q2 P(pos(k),1)];
          end
          k=k-1;
        end
      end
    end
end

x=X(:,Q);
xs=X_test(:,Q);

for i=1:length(Q)
    z=unique(x(:,i));
    for j=length(z):-1:1
        x(x(:,i)==z(j),i)=j-1;
        xs(xs(:,i)==z(j),i)=j-1;
    end
end

y=Y';
x=x+2;
ys=Y_test';
xs=xs+2;
save(['baseline1part0_'num2str(out)'prueba_new.mat'],'x','y','xs','ys','Q','Q1','Q2');

%genera el archivo baseline1part0_1prueba_new.mat
```

A.5 baseline1part1.m

```
function baseline1part1(out,N,factor)

%out es la salida. Tiene un valor fijo de 1 (Intento de suicidio)
%N son las variables con mayor información mutua con la salida que se van a considerar para
%obtener el mejor cuarteto (10, 20, 30)
%factor: es la variable por la cual se multiplica los casos de pacientes que han tenido al menos, un
%intento de suicidio.
```

```
%Se obtienen los 100 mejores conjuntos de variables. Para ordenarlas según la información mutua,
%se utiliza el programa baseline1part0.m

out=1;
factor = 4;
N=30;
load(['baseline1part0_' num2str(out) 'prueba_new.mat']);

c1=100; %Cuantos de los mejores cuartetos se van a coger
N=min(N,size(x,2));

Err=zeros(N,N,N,N);
Err2=zeros(N,N,N,N);    %ND: No Detección
Err3=zeros(N,N,N,N);    %FA: Falsa Alarma
for i1=1:N-3
  for i2=i1+1:N-2
    mul(1)=length(unique(x(:,i2)));
    for i3=i2+1:N-1
      mul(2)=length(unique(x(:,i3)));
      for i4=i3+1:N
        mul(3)=length(unique(x(:,i4)));
        aux=mul(1)*mul(2)*mul(3)*x(:,i1)+mul(2)*mul(3)*x(:,i2)+mul(3)*x(:,i3)+x(:,i4);
        S=unique(aux);
        for k=1:length(S)
          v1=length(find(y==1 & aux==S(k)));
          v0=length(find(y==0 & aux==S(k)));
          if((v1-1)>=factor*v0)
            Err(i1,i2,i3,i4)=Err(i1,i2,i3,i4)+factor*v0; %Err= FA+factor*ND
            Err2(i1,i2,i3,i4)=Err2(i1,i2,i3,i4)+v0;
          elseif(v1<factor*(v0-1))
            Err(i1,i2,i3,i4)=Err(i1,i2,i3,i4)+v1;
            Err3(i1,i2,i3,i4)=Err3(i1,i2,i3,i4)+v1;
          else
            Err(i1,i2,i3,i4)=Err(i1,i2,i3,i4)+v1+factor*v0;
            Err3(i1,i2,i3,i4)=Err3(i1,i2,i3,i4)+v1;
            Err2(i1,i2,i3,i4)=Err2(i1,i2,i3,i4)+v0;
          end
        end
      end
```

```
    end
  end
end

Err(Err==0)=10000;
R=zeros(c1,3+4);
for i=1:c1
  [val,pos1]=min(Err);
  [val,pos2]=min(val);
  [val,pos3]=min(val);
  [val,pos4]=min(val);
  R(i,1:7)=[val
Err2(pos1(1,pos2(1,1,pos3(pos4),pos4),pos3(pos4),pos4),pos2(1,1,pos3(pos4),pos4),pos3(pos4),pos4)
Err3(pos1(1,pos2(1,1,pos3(pos4),pos4),pos3(pos4),pos4),pos2(1,1,pos3(pos4),pos4),pos3(pos4),pos4)
pos1(1,pos2(1,1,pos3(pos4),pos4),pos3(pos4),pos4) pos2(1,1,pos3(pos4),pos4) pos3(pos4) pos4];

Err(pos1(1,pos2(1,1,pos3(pos4),pos4),pos3(pos4),pos4),pos2(1,1,pos3(pos4),pos4),pos3(pos4),po
s4)=10000;
end
save(['baseline1part1_' num2str(out) '_' num2str(N)  '_' num2str(factor) '.mat'],'R');
```

A.6 baseline1part2.m

Junta todos los clasificadores obtenidos para los distintos N utilizados en baseline1part1.m.

```
function baseline1part2(out,N,factor)

load(['baseline1part1_' num2str(out) '_' num2str(N(1)) '_' num2str(factor) '.mat']);
RT=R;
for i=2:length(N)
  load(['baseline1part1_' num2str(out) '_' num2str(N(i)) '_' num2str(factor) '.mat']);
  for j=1:size(R,1)
    in=find(RT(:,1)==R(j,1));
    if(length(in))
      if(length(find(sum(abs(ones(length(in),1)*R(j,:)-RT(in,:)),2)==0))==0)
        RT=[RT;R(j,:)];
```

```
        end
      else
        RT=[RT;R(j,:)];
      end
    end
end
R=RT;
save(['baseline1part2_' num2str(out) '_' num2str(factor) '.mat'],'R');
```

A.7 baseline1part3.m

```
function baseline1part3(out,N1,c1,factor)
%out es la salida (out=1)
%N1 = variables a considerar
%c1 indica la posición de las variables (grupo de 100)

c2=10; %Número de variables que se añaden al conjunto de 4 que ya se tenía
load(['baseline1part0_' num2str(out) 'prueba_new.mat']);
load(['baseline1part2_' num2str(out) '_' num2str(factor) '.mat']);

N1=min(N1,size(x,2));
R2=zeros(c2+1,c2+7);
R2(1,1:7)=R(c1,1:7);
clear R
RR=R2(1,4:7);
W=setdiff(1:N1,RR);
for i=1:c2
  i
  Err_aux=zeros(1,length(W));
  Err_aux2=zeros(1,length(W));
  Err_aux3=zeros(1,length(W));
  for l=1:length(W)
    aux=x(:,RR(1));
    for k=2:length(RR)
      aux=length(unique(x(:,RR(k))))*aux+x(:,RR(k));
    end
    aux=length(unique(x(:,W(l))))*aux+x(:,W(l));
    S=unique(aux);
```

```
    for k=1:length(S)
      v1=length(find(y==1 & aux==S(k)));
      v0=length(find(y==0 & aux==S(k)));
      if((v1-1)>=factor*v0)
        Err_aux(l)=Err_aux(l)+factor*v0;
        Err_aux2(l)=Err_aux2(l)+v0;
      elseif(v1<factor*(v0-1))
        Err_aux(l)=Err_aux(l)+v1;
        Err_aux3(l)=Err_aux3(l)+v1;
      else
        Err_aux(l)=Err_aux(l)+v1+factor*v0;
        Err_aux3(l)=Err_aux3(l)+v1;
        Err_aux2(l)=Err_aux2(l)+v0;
      end
    end
  end
  [val,pos]=min(Err_aux);
  RR=[RR W(pos)];
  R2(i+1,1:7+i)=[val Err_aux2(pos) Err_aux3(pos) RR];
  R2(i+1,1:7+i)
  W=setdiff(W,W(pos));
end

save(['baseline1part3_' num2str(out) '_' num2str(N1) '_' num2str(c1) '_' num2str(factor)
'.mat'],'R2');
```

A.8 main_R2_2.m

Ejecutable del programa baseline1part3.m. Pinta las gráficas de probabilidades de No detección y Falsa alarma y para cada valor de Factor, los clasificadores aparecen de un color determinado.

```
%pinta la curva de datos obtenidos en baseline1part3.m
%columna 3 FA (eje X), columna 2 ND (eje y)
% n para representar N1 (50,100,500,1000)
% j para representar c1(desde 1 hasta 100)
% f para representar el factor (1,2,3,4,5,8,16,32,64)
```

```
%creo una matriz de 2x3600 para almacenar las columnas 2 y 3
Matriz=ones(3600,18);
%almaceno la matriz R2
c=ones(11,17);
%actualizo los índices de M
r=0;
%valor de N1
n=1000;
   %recorremos los valores de c1
   for j =1:100


      %recorremos los valores de factor
      factor=[1, 2, 3, 4, 5, 8, 16, 32, 64];
      for z=1:9
       f=factor(z);
         %lee el fichero correspondiente a esos valores

load(['baseline1part3_' num2str(1) '_' num2str(n) '_' num2str(j) '_' num2str(f) '.mat'],'R2');


%almaceno en M
   c=R2(:,:);
   for p=1:11
      for a=1:17
      s=p+r;
   Matriz(s,a)=c(p,a);
      end
   if f == 1
   Matriz(s,18)=1;
   end
    if f == 2
   Matriz(s,18)=2;
    end
    if f == 3
   Matriz(s,18)=3;
    end
    if f == 4
   Matriz(s,18)=4;
    end
    if f == 5
```

```
Matriz(s,18)=5;
 end
 if f == 8
Matriz(s,18)=8;
 end
 if f == 16
Matriz(s,18)=16;
 end
 if f == 32
Matriz(s,18)=32;
 end
 if f == 64
Matriz(s,18)=64;
 end
end
   r=r+11;

%dibujar normalizando
 if f == 1
 plot(R2(:,3)/9217, 1-(R2(:,2)/783), '*c');
 hold on
 end
 if f == 2
 plot(R2(:,3)/9217, 1-(R2(:,2)/783), '*r');
 hold on
 end
 if f == 3
 plot(R2(:,3)/9217, 1-(R2(:,2)/783), '*');
 hold on
 end
 if f == 4
 plot(R2(:,3)/9217, 1-(R2(:,2)/783), '.y');
 hold on
 end
 if f == 5
 plot(R2(:,3)/9217, 1-(R2(:,2)/783), '.g');
 hold on
 end
 if f == 8
```

```
   plot(R2(:,3)/9217, 1-(R2(:,2)/783), '.');
   hold on
    end
    if f == 16
   plot(R2(:,3)/9217, 1-(R2(:,2)/783), '.m');
   hold on
    end
    if f == 32
   plot(R2(:,3)/9217, 1-(R2(:,2)/783), '.r');
   hold on
    end
    if f == 64
   plot(R2(:,3)/9217, 1-(R2(:,2)/783), '.b');
   hold on
   end
 end
end
save ('Matriz');
```

A.9 sens_spec.m

La ejecución de este programa muestra la gráfica con los clasificadores generados, permite seleccionar cualquiera de ellos y muestra las variables que lo componen, así como las probabilidades de No detección, Falsa alarma, Sensibilidad, Especificidad y el valor de Factor.

```
clear
close all

load Matriz
load baseline1part0_1_new.mat

%figure(1)
III=find(Matriz(:,3)/9217<0.25 & Matriz(:,3)/9217>0.15);

H=plot(Matriz(:,3)/9217,(783-Matriz(:,2))/783,'.',[1 0],[0 1]);
disp(' ')
disp(' ')
```

```
disp('Reescale the figure is needed')
disp(' ')
disp(' ')
pause

disp(' ')
disp(' ')
disp('Select a point from the Figure')
disp('to see which are the variables involved.')
disp(' ')
disp(' ')
[h,v]=ginput(1);
[val2,pos2]=min((Matriz(:,3)/9217-h).^2+((783-Matriz(:,2))/783-v).^2);
E=Matriz(pos2,:);
E=E(find(E));
fprintf(1,'\nProbability of False Alarm: %1.3f',Matriz(pos2,3)/9217)
fprintf(1,'\nProbability of Misdetection: %1.3f',Matriz(pos2,2)/783)
fprintf(1,'\nSpecificity: %1.3f',(9217-Matriz(pos2,3))/9217)
fprintf(1,'\nSensitivity: %1.3f\n',(783-Matriz(pos2,2))/783)
fprintf(1,'\nImportance Factor: %d\n',Matriz(pos2, 18))

features1
close all
[num2str(Q2(E(5:end))') Features1(Q2(E(5:end)),:) Features2(Q2(E(5:end)),:)];

save(['Modelo' num2str(randint(1,1,1e8)) '.mat'],'E');
```

Referencias

[1] MITCHELL, Tom. Machine Learning. Editorial McGraw-Hill, 1997. ISBN: 0070428077

[2] BORRAJO MILLÁN, Daniel; GONZÁLEZ BOTICARIO, Jesús; ISASI VIÑUELA, Pedro. Aprendizaje automático. Editorial Sanz y Torres, 2006. ISBN: 8496094731, 9788496094734

[3] Asignatura *Tratamiento Digital de la Información*, Universidad Carlos III de Madrid. Curso 2010/2011.

[4] National Institutes of Health. Alcohol use and alcohol use disorders in the United States, a 3-year follow-up: main findings from the2004-2005 wave 2 national epidemiologic survey on alcohol and related conditions (NESARC). Manual de referencia de datos, Vol.8, num 2. Septiembre de 2010. Disponible en: <http://pubs.niaaa.nih.gov/publications/NESARC_DRM2/NESARC2DRM.pdf>

[5] HERNÁNDEZ ORALLO, José. Técnicas de minería de datos. Universidad Politécnica de Valencia. Disponible en: <http://users.dsic.upv.es/~jorallo/master/dm3.pdf>

[6] GUTIERREZ OSUNA, Ricardo. Intelligent Sensor Systems. Wright State University. Disponible en: <http://courses.cs.tamu.edu/rgutier/ceg499_s02/l13.pdf>

[7] COSMING GONZÁLEZ, Carlos Andrés. Pronóstico de captura de anchovetas de la zona norte de Chile usando vector de soporte autorregresivo. Director: Nibaldo Rodríguez Agurto. Pontificia Universidad Católica de Valparaíso. Escuela de Ingeniería Informática. Abril 2012. Disponible en: <http://opac.ucv.cl/pucv_txt/pucv/Txt-1500/UCF1571_01.pdf>

[8] MOORE, Andrew W. Cross-validation for detecting and preventing overfitting. School of Computer Science. Carnegie Mellon University. Disponible en: <http://www.autonlab.org/tutorials/overfit10.pdf>

[9]FERNÁNDEZ REBOLLO, Fernando; BORRAJO MILLÁN, Daniel. Aprendizaje automático. Grupo de Planificación y Aprendizaje (PLG). Escuela Politécnica Superior. Universidad Carlos III de Madrid. Disponible en: <http://ocw.uc3m.es/ingenieria-informatica/aprendizaje-automatico/aa-ocw-contenido.pdf>

[10] GIRÁLDEZ ROJO, Raúl. Mejoras en eficiencia y eficacia de algoritmos evolutivos para aprendizaje supervisado. Directores: Dr. D. José C. Riquelme Santos; Dr. D. Jesús S. Aguilar Ruiz. Departamento de Lenguajes y sistemas informáticos. Universidad de Sevilla. Septiembre 2003. Disponible en: <http://www.lsi.us.es/docs/doctorado/memorias/Memoria-Raul-Giraldez.pdf>

[11] BURGOS, Andrés C. Selección de variables en problemas multiclase. Director: Dr. Pablo M. Granitto. Facultad de Ciencias Exactas, Ingeniería y Agrimensura. Universidad Nacional de Rosario. Marzo 2009. Disponible en: <http://www.fceia.unr.edu.ar/lcc/t523/uploads/12.pdf>